IP AS THE IOT NETWORK LAYER

Introduction

- Concepts Covered:
  - Discussion on Network layer connectivity, or referred **Layer 3**
  - **The Business Case for IP:** the advantages of IP from an IoT perspective and introduces the concepts of adoption and adaptation.
  - **The Need for Optimization:** the challenges of constrained nodes and devices when deploying IP along with migration from IPv4 to IPv6 and how it affects IoT networks.
  - **Optimizing IP for IoT:** the common protocols and technologies in IoT networks utilizing IP,including 6LoWPAN, 6TiSCH, and RPL.
  - **Profiles and Compliances:** some of the most significant organizations and standards bodies involved with IP connectivity and IoT.

The Business Case for IP

- Data flowing from or to “things” is consumed, controlled, or monitored by data center servers either in the cloud or in locations that may be distributed or centralized.
- The system solutions combining various physical and data link layers call for an architectural approach with a common layer(s) independent from the lower (connectivity) and/or upper (application) layers. **That is using IP (Internet Protocol)**

The Business Case for IP

The Key Advantages of Internet Protocol

- In information technology (IT) or operational technology (OT), the lifetime of the underlying technologies and products.
- To guarantee multi-year lifetimes is to define a layered architecture such as the 30-year-old IP architecture.

The Business Case for IP

The Key Advantages of Internet Protocol

- **The key advantages of the IP for the Internet of Things**:
  - Open and standards-based
  - Versatile
  - Ubiquitous
  - Scalable
  - Manageable and highly secure
  - Stable and resilient
  - Consumers' market adoption
  - The innovation factor

The Business Case for IP

The Key Advantages of Internet Protocol

- **Open and standards-based**
  - The Internet of Things creates a new paradigm in which devices, applications, and users can leverage a large set of devices and functionalities while guaranteeing interchangeability and interoperability, security, and management.
  - This calls for implementation, validation, and deployment of open, standards-based solutions.
  - While many standards development organizations (SDOs) are working on Internet of Things definitions, frameworks, applications, and technologies, none are questioning the role of the Internet Engineering Task Force (IETF) as the foundation for specifying and optimizing the network and transport layers.
  - The IETF is an open standards body that focuses on the development of the Internet Protocol suite and related Internet technologies and protocols.

The Business Case for IP

The Key Advantages of Internet Protocol

- **Versatile:**
  - A large spectrum of access technologies is available to offer connectivity of "things" in the last mile.
  - Even if physical and data link layers such as Ethernet, Wi-Fi, and cellular are widely adopted, the history of data communications demonstrates that no given wired or wireless technology fits all deployment criteria.
  - Communication technologies evolve at a pace faster than the expected 10- to 20-year lifetime of OT solutions.
  - So, the layered IP architecture is well equipped to cope with any type of physical and data link layers.
  - This makes IP ideal as a long-term investment because various protocols at these layers can be used in a deployment now and over time, without requiring changes to the whole solution architecture and data flow.

The Business Case for IP

The Key Advantages of Internet Protocol

- **Ubiquitous:**
  - All recent operating system releases, from general-purpose computers and servers to lightweight embedded systems (TinyOS, Contiki, and so on), have an integrated dual (IPv4 and IPv6) IP stack that gets enhanced over time.
  - IoT application protocols in many industrial OT solutions have been updated in recent years to run over IP.
  - While these updates have mostly consisted of IPv4 to this point, recent standardization efforts in several areas are adding IPv6.
  - In fact, IP is the most pervasive protocol when you look at what is supported across the various IoT solutions and industry verticals.

The Business Case for IP

The Key Advantages of Internet Protocol

- **Scalable:**
  - As the common protocol of the Internet, IP has been massively deployed and tested for robust scalability.
  - Millions of private and public IP infrastructure nodes have been operational for years, offering strong foundations for those not familiar with IP network management.
  - Adding huge numbers of "things" to private and public infrastructures may require optimizations and design rules specific to the new devices.
  - However, you should realize that this is not very different from the recent evolution of voice and video endpoints integrated over IP.
  - IP has proven before that scalability is one of its strengths.

The Business Case for IP

The Key Advantages of Internet Protocol

- **Manageable and highly secure:**
  - Communications infrastructure require appropriate management and security capabilities for proper operations.
  - One of the benefits that comes from 30 years of operational IP networks is the well understood network management and security protocols, mechanisms, and toolsets that are widely available.
  - Adopting IP network management also brings an operational business application to OT. Well-known network and security management tools are easily leveraged with an IP network layer.
  - However, you should be aware that despite the secure nature of IP, real challenges exist in this area.
  - Specifically, the industry is challenged in securing constrained nodes, handling legacy OT protocols, and scaling operations.

The Business Case for IP

The Key Advantages of Internet Protocol

- **Stable and resilient:**
  - IP has been around for 30 years, and it is clear that IP is a workable solution.
  - IP has a large and well-established knowledge base and, more importantly, it has been used for years in critical infrastructures, such as financial and defense networks.
  - In addition, IP has been deployed for critical services, such as voice and video, which have already transitioned from closed environments to open IP standards.
  - Finally, its stability and resiliency benefit from the large ecosystem of IT professionals who can help design, deploy, and operate IP-based solutions.

The Business Case for IP

The Key Advantages of Internet Protocol

- **Consumers' market adoption:**
  - When developing IoT solutions and products targeting the consumer market, vendors know that consumers' access to applications and devices will occur predominantly over broadband and mobile wireless infrastructure.
  - The main consumer devices range from smart phones to tablets and PCs. The common protocol that links IoT in the consumer space to these devices is IP.

The Business Case for IP

The Key Advantages of Internet Protocol

- **The innovation factor:**
  - The past two decades have largely established the adoption of IP as a factor for increased innovation.
  - IP is the underlying protocol for applications ranging from file transfer and e-mail to the World Wide Web, ecommerce, social networking, mobility, and more.
  - Even the recent computing evolution from PC to mobile and mainframes to cloud services are perfect demonstrations of the innovative ground enabled by IP.
  - Innovations in IoT can also leverage an IP underpinning.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- The use of numerous network layer protocols in addition to IP is often a point of contention between computer networking experts. Typically, one of two models, adaptation or adoption, is proposed:
  - Adaptation means application layered gateways (ALGs) must be implemented to ensure the translation between non-IP and IP layers.
  - Adoption involves replacing all non-IP layers with their IP layer counterparts, simplifying the deployment model and operations.
- How to implement IP in data center, cloud services, and operation centers hosting IoT applications.
- Adoption of IP is more complicated and often makes running IP end-to-end more difficult.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- In the industrial and manufacturing sector, Solutions and product lifecycles many protocols have been developed for serial communications.
  - While IP and Ethernet support were not specified in the initial versions, more recent specifications for these serial communications protocols integrate Ethernet and IPv4.
- Supervisory control and data acquisition (SCADA) applications that operate both the IP adaptation model and the adoption model.
  - Implementations that make use of IP adaptation have SCADA devices attached through serial interfaces to a gateway tunneling or translating the traffic.
  - With the IP adoption model, SCADA devices are attached via Ethernet to switches and routers forwarding their IPv4 traffic.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- ZigBee that runs a non-IP stack between devices and a ZigBee gateway that forwards traffic to an application server.
  - A ZigBee gateway often acts as a translator between the ZigBee and IP protocol stacks.

- Following factors when trying to determine which model is best suited for IP connectivity:
  - Bidirectional versus unidirectional data flow
  - Overhead for last-mile communications paths
  - Data flow model
  - Network diversity

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- Bidirectional versus unidirectional data flow
  - While bidirectional communications are generally expected, some last-mile technologies offer optimization for unidirectional communication.
  - e.g. RFC 7228, may only infrequently need to report a few bytes of data to an application LPWA technologies, include fire alarms sending alerts or daily test reports, electrical switches being pushed on or off, and water or gas meters sending weekly indexes.
- For these cases, it is not necessarily worth implementing a full IP stack.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **Overhead for last-mile communications paths:**
  - IP adoption implies a layered architecture with a per-packet overhead that varies depending on the IP version.
    - IPv4 has 20 bytes, IPv6 has 40 bytes, UDP has 8 bytes and TCP has a minimum of 20 bytes
  - If the data to be forwarded by a device is infrequent and only a few bytes, you can potentially have more header overhead than device data again, (in the case of LPWA technologies).
    - Consequently, you need to decide whether the IP adoption model is necessary and, if it is, how it can be optimized.
    - This same consideration applies to control plane traffic that is run over IP for low-bandwidth, last-mile links.
    - Routing protocol and other verbose network services may either not be required or call for optimization.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **Data flow model:**
  - One benefit of the IP adoption model is the end-to-end nature of communications.
    - Any node can easily exchange data with any other node in a network, although security, privacy, and other factors may put controls and limits on the "end-to-end" concept.
  - However, in many IoT solutions,
    - a device's data flow is limited to one or two applications.
    - In this case, the adaptation model can work because translation of traffic needs to occur only between the end device and one or two application servers.
  - Depending on the network topology and the data flow needed, both IP adaptation and adoption models have roles to play in last-mile connectivity.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **Network diversity:**
  - One of the drawbacks of the adaptation model is a general dependency on single PHY and MAC layers.
    - For example, ZigBee devices must only be deployed in ZigBee network islands.
  - Therefore, a deployment must consider which applications have to run on the gateway connecting these islands and the rest of the world.
  - Integration and coexistence of new physical and MAC layers or new applications impact how deployment and operations have to be planned.
  - This is not a relevant consideration for the adoption model.

The Need for Optimization

- Internet of Things will largely be built on the Internet Protocol suite
- In coping with the integration of non-IP devices, may need to deal with the limits at the device and network levels that IoT often imposes.
- Therefore, optimizations are needed at various layers of the IP stack to handle the restrictions that are present in IoT networks.
- The following concepts take a detailed look at why optimization is necessary for IP.
  - Constrained Nodes
  - Constrained Networks
  - IP Versions

The Need for Optimization: Constrained Nodes

- IoT having different classes of devices coexist.
- Depending on its functions in a network, a "thing" architecture may or may not offer similar characteristics compared to a generic PC or server in an IT environment.
- Another limit is that this network protocol stack on an IoT node may be required to communicate through an unreliable path.
  - Even if a full IP stack is available on the node, this causes problems such as limited or unpredictable throughput and low convergence when a topology change occurs.
- Power consumption is a key characteristic of constrained nodes.
  - Battery Enabled with life span of Months to 10 years
  - Battery-powered nodes impact communication intervals.
  - The Node one that is "always on" instead another option is "always off," which means communications are enabled only when needed to send data.

The Need for Optimization Constrained Nodes

- IoT constrained nodes can be classified as follows:
  - **Devices that are very constrained in resources, may communicate infrequently to transmit a few bytes, and may have limited security and management capabilities:** This drives the need for the IP adaptation model, where nodes communicate through gateways and proxies.
  - **Devices with enough power and capacities to implement a stripped- down IP stack or non-IP stack:** In this case, you may implement either an optimized IP stack and directly communicate with application servers (adoption model) or go for an IP or non-IP stack and communicate through gateways and proxies (adaptation model).
  - **Devices that are similar to generic PCs in terms of computing and power resources but have constrained networking capacities, such as bandwidth:** These nodes usually implement a full IP stack (adoption model), but network design and application behaviors must cope with the bandwidth constraints.

The Need for Optimization Constrained Nodes

- In constrained nodes, the costs of computing power, memory, storage resources, and power consumption are generally decreasing.
- At the same time, networking technologies continue to improve and offer more bandwidth and reliability.

The Need for Optimization Constrained Networks

- Low-speed connections (like low-speed modems) demonstrated that IP could run over low-bandwidth networks.
- High-speed connections are not usable by some IoT devices in the last mile.
  - The reasons include the implementation of technologies with low bandwidth, limited distance and bandwidth due to regulated transmit power, and lack of or limited network services.
  - A constrained network can have high latency and a high potential for packet loss.
  - Constrained networks are often referred to as low-power and lossy networks (LLNs).
  - Constrained networks operate between a few kbps and a few hundred kbps and may utilize a star, mesh, or combined network topologies, ensuring proper operations.

The Need for Optimization Constrained Networks

- In constrained network, it is not unusual for the packet delivery rate (PDR) to oscillate between low and high percentages.
- Large bursts of unpredictable errors and even loss of connectivity at times may occur, where packet delivery variation may fluctuate greatly during the course of a day.
- Latency and control plane reactivity:
  - One of the golden rules in a constrained network is to "underreact to failure."
  - Due to the low bandwidth, a constrained network that overreacts can lead to a network collapse which makes the existing problem worse.
- Control plane traffic must also be kept at a minimum; otherwise, it consumes the bandwidth that is needed by the data traffic.
- The power consumption in battery-powered nodes:
  - Any failure or verbose control plane protocol may reduce the lifetime of the batteries.
- This led to work on optimizing protocols for IoT

The Need for Optimization IP Versions

- IETF has been working on transitioning the Internet from IP version 4 to IP version 6.
- The main driving force has been the lack of address space in IPv4 as the Internet has grown.
- IPv6 has a much larger range of addresses that should not be exhausted for the foreseeable future.
- Today, both versions of IP run over the Internet, but most traffic is still IPv4 based.
- Internet of Things has the Internet itself and support both IPv4 and IPv6 versions concurrently.
  - Techniques such as tunneling and translation need to be employed in IoT solutions to ensure interoperability between IPv4 and IPv6.

The Need for Optimization IP Versions

- The following are some of the main factors applicable to IPv4 and IPv6 support in an IoT solution:
  - Application Protocol
  - Cellular Provider and Technology
  - Serial Communications
  - IPv6 Adaptation Layer

- **Application Protocol:** IoT devices implementing Ethernet or Wi-Fi interfaces can communicate over both IPv4 and IPv6, but the application protocol may dictate the choice of the IP version.
  - For example, SCADA protocols such as DNP3/IP (IEEE 1815), Modbus TCP, or the IEC 60870-5-104 standards are specified only for IPv4.
  - So, there are no known production implementations by vendors of these protocols over IPv6 today.
  - For IoT devices with application protocols defined by the IETF, such as HTTP/HTTPS, CoAP, MQTT, and XMPP, both IP versions are supported.
  - The selection of the IP version is only dependent on the implementation.

The Need for Optimization IP Versions

- **Cellular Provider and Technology:** IoT devices with cellular modems are dependent on the generation of the cellular technology as well as the data services offered by the provider.
  - For the first three generations of data services GPRS, Edge, and 3G IPv4 is the base protocol version.
  - Consequently, if IPv6 is used with these generations, it must be tunneled over IPv4.
  - On 4G/LTE networks, data services can use IPv4 or IPv6 as a base protocol, depending on the provider.

The Need for Optimization IP Versions

- **Serial Communications:**
  - Data is transferred using either proprietary or standards-based protocols, such as DNP3, Modbus, or IEC 60870-5-101.
  - In the past, communicating this serial data over any sort of distance could be handled by an analog modem connection.
    - However, as service provider support for analog line services has declined, the solution for communicating with these legacy devices has been to use local connections. To make this work, you connect the serial port of the legacy device to a nearby serial port on a piece of communications equipment, typically a router. This local router then forwards the serial traffic over IP to the central server for processing.
    - Encapsulation of serial protocols over IP leverages mechanisms such as raw socket TCP or UDP. While raw socket sessions can run over both IPv4 and IPv6, current implementations are mostly available for IPv4 only.

The Need for Optimization IP Versions

- **IPv6 Adaptation Layer:**
- IPv6-only adaptation layers for some physical and data link layers for recently standardized IoT protocols support only IPv6.
- While the most common physical and data link layers (Ethernet, Wi-Fi, and so on) stipulate adaptation layers for both versions, newer technologies,
  - such as IEEE 802.15.4 (Wireless Personal Area Network), IEEE 1901.2, and ITU G.9903 (Narrowband Power Line Communications) only have an IPv6 adaptation layer specified.
- This means that any device implementing a technology that requires an IPv6 adaptation layer must communicate over an IPv6-only subnetwork.
  - This is reinforced by the IETF routing protocol for LLNs, RPL, which is IPv6 only.

Optimizing IP for IoT

- While the Internet Protocol is key for a successful Internet of Things, constrained nodes and constrained networks mandate optimization at various layers and on multiple protocols of the IP architecture

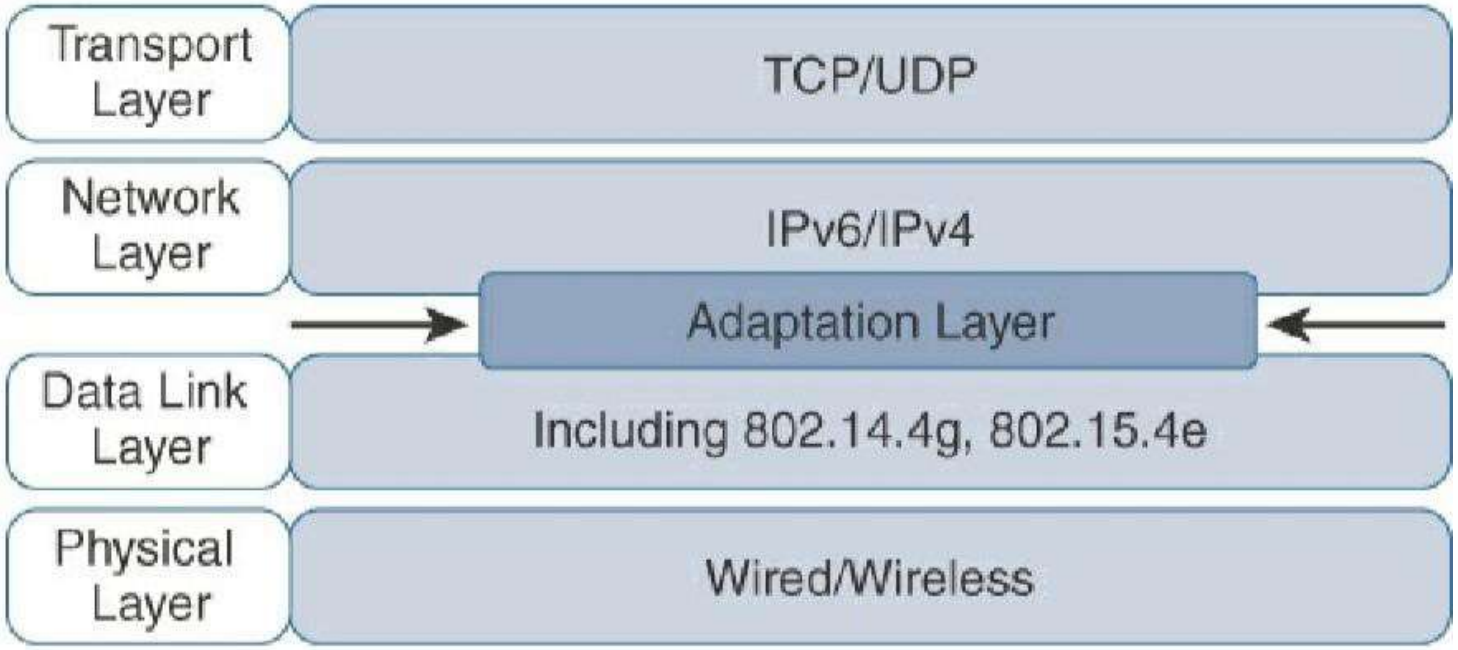


Figure 5-1 Optimizing IP for IoT Using an Adaptation Layer

Optimizing IP for IoT

- **The following optimizations technique of IP already available**:
  - From 6LoWPAN to 6Lo
    - Header Compression
    - Fragmentation
    - Mesh Addressing
    - Mesh-Under Versus Mesh-Over Routing
    - 6Lo Working Group
  - 6TiSCH
  - RPL
    - Objective Function (OF)
    - Rank
    - RPL Headers
    - Metrics
  - Authentication and Encryption on Constrained Nodes
    - ACE
    - DICE

Optimizing IP for IoT From 6LoWPAN to 6Lo

- In the IP architecture, the transport of IP packets over any given Layer 1 (PHY) and Layer 2 (MAC) protocol must be defined.
- The initial focus of the 6LoWPAN working group was to optimize the transmission of IPv6 packets over constrained networks such as IEEE 802.15.4.

Optimizing IP for IoT From 6LoWPAN to 6Lo

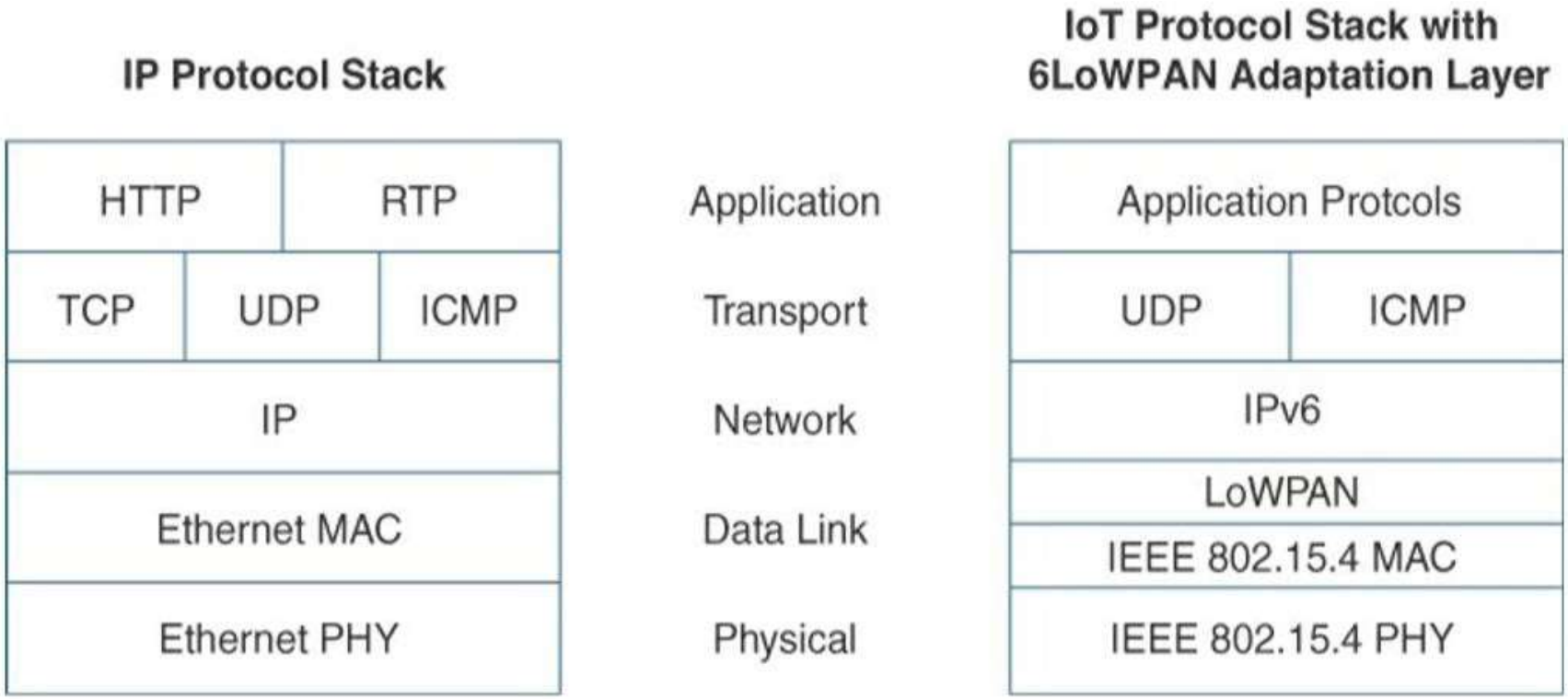


Figure 5-2 Comparison of an IoT Protocol Stack Utilizing 6LoWPAN and an IP Protocol Stack

Optimizing IP for IoT From 6LoWPAN to 6Lo

- The 6LoWPAN working group published several RFCs (Request for Comments by IETF), but RFC defines frame headers for the capabilities of
  - Header compression,
  - Fragmentation,
  - Mesh addressing.

Optimizing IP for IoT From 6LoWPAN to 6Lo

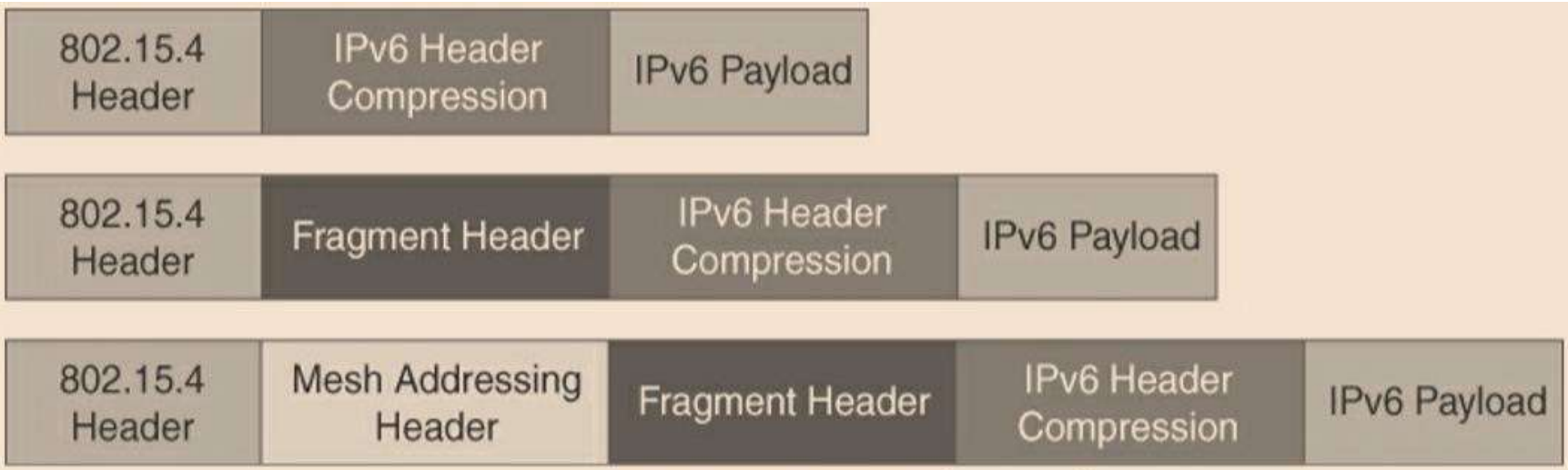


Figure 5-3 6LoWPAN Header Stacks

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **Header Compression:**
  - Shrinks the size of IPv6's 40-byte headers and User Datagram Protocol's (UDP's) 8-byte headers down as low as 6 bytes combined in some cases.
  - Header compression for 6LoWPAN is only defined for an IPv6 header and not for IPv4.
    - However, a number of factors affect the amount of compression, such as implementation of RFC, whether UDP is included, and various IPv6 addressing scenarios.
  - 6LoWPAN works by taking advantage of shared information known by all nodes from their participation in the local network.
  - In addition, it omits some standard header fields by assuming commonly used values.

Optimizing IP for IoT From 6LoWPAN to 6Lo

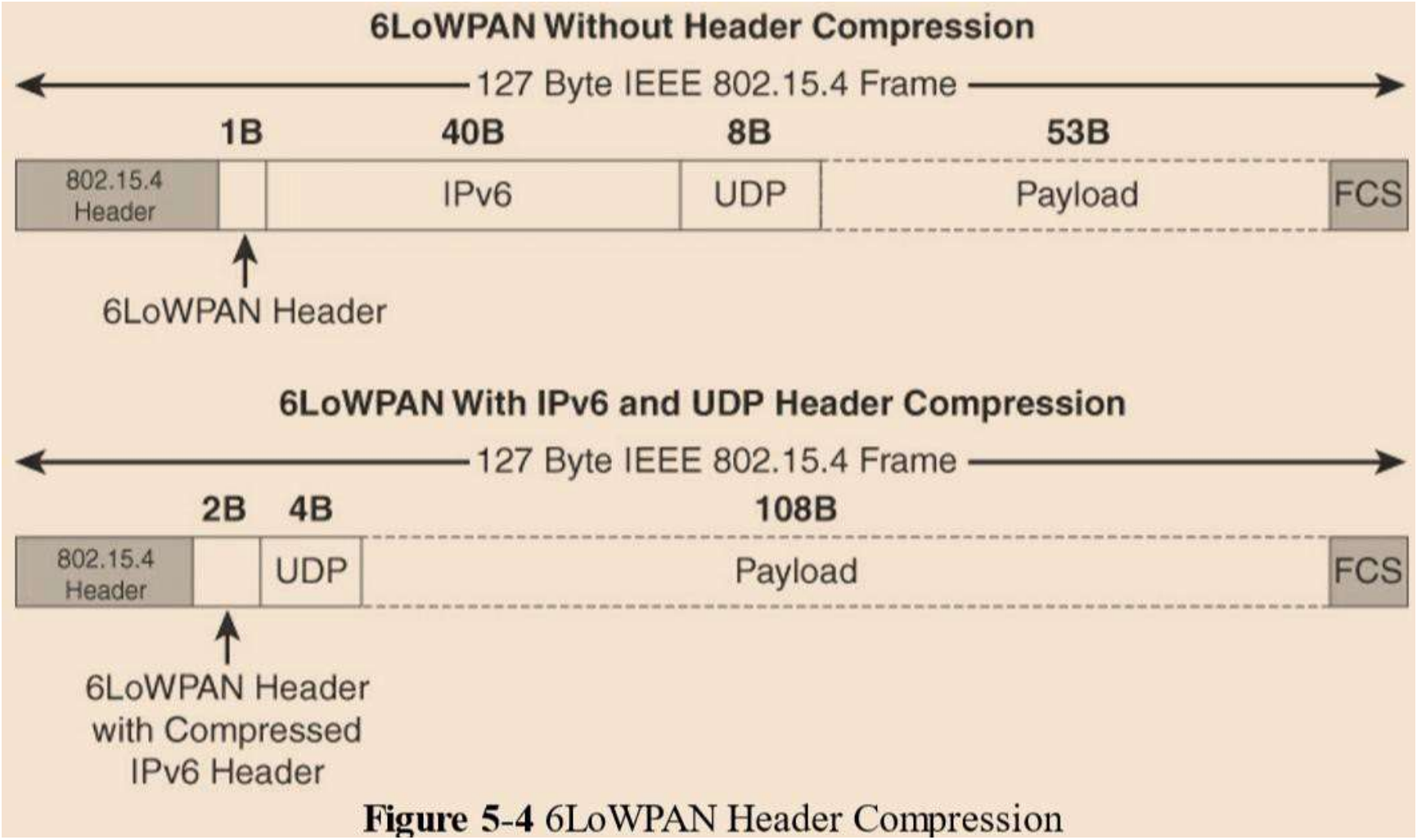


Figure 5-4 6LoWPAN Header Compression

Optimizing IP for IoT From 6LoWPAN to 6Lo

- At the top of, a 6LoWPAN frame without any header compression enabled:
  - The full 40-byte IPv6 header and 8-byte UDP header are visible.
  - The 6LoWPAN header is only a single byte in this case.
  - Uncompressed IPv6 and UDP headers leave only 53 bytes of data payload out of the 127-byte maximum frame size in the case of IEEE 802.15.4.
- The bottom half of shows a frame where header compression enabled:
  - The 6LoWPAN header increases to 2 bytes to accommodate the compressed IPv6 header, and UDP has been reduced in half, to 4 bytes from 8.
  - Most importantly, the header compression has allowed the payload to more than double, from 53 bytes to 108 bytes, which is obviously much more efficient.

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **Fragmentation:**
  - The maximum transmission unit (MTU) for an IPv6 network must be at least 1280 bytes.
  - The term MTU defines the size of the largest protocol data unit that can be passed.
  - For IEEE 802.15.4, 127 bytes is the MTU.
  - A problem because of IPv6, with a much larger MTU, is carried inside the 802.15.4 frame with a much smaller one.
  - To remedy this situation, large IPv6 packets must be fragmented across multiple 802.15.4 frames at Layer 2.

Optimizing IP for IoT From 6LoWPAN to 6Lo

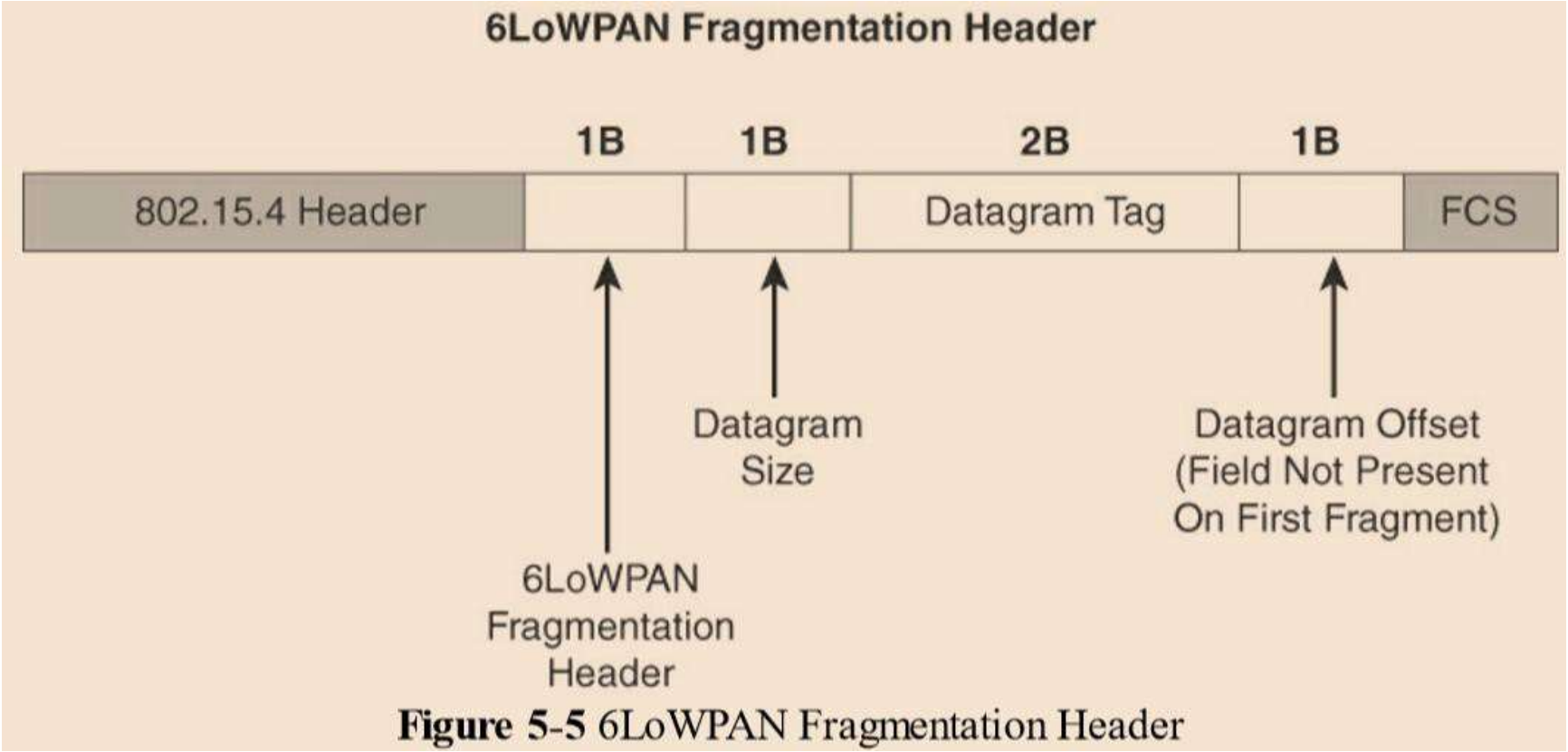


Figure 5-5 6LoWPAN Fragmentation Header

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **The fragment header utilized by 6LoWPAN is composed of three primary fields:**
  - Datagram Size: The 1-byte field specifies the total size of the unfragmented payload
  - Datagram Tag: identifies the set of fragments for a payload.
  - Datagram Offset: field delineates how far into a payload a particular fragment occurs.
- **The 6LoWPAN fragmentation header field itself uses a unique bit value to identify that the subsequent fields behind it are fragment fields as opposed to another capability, such as header compression.**
- **In the first fragment, the Datagram Offset field is not present because it would simply be set to 0.**
- **This results in the first fragmentation header for an IPv6 payload being only 4 bytes long.**

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **Mesh Addressing:**
  - The purpose of the 6LoWPAN mesh addressing function is to forward packets over multiple hops.
  - Three fields are defined for this header:
    - Hop Limit: The hop limit for mesh addressing also provides an upper limit on how many times the frame can be forwarded. Each hop decrements this value by 1 as it is forwarded. Once the value hits 0, it is dropped and no longer forwarded.
    - Source Address, and Destination Address: The Source Address and Destination Addressfields for mesh addressing are IEEE 802.15.4 addresses indicating the endpoints of an IP hop.

Optimizing IP for IoT From 6LoWPAN to 6Lo

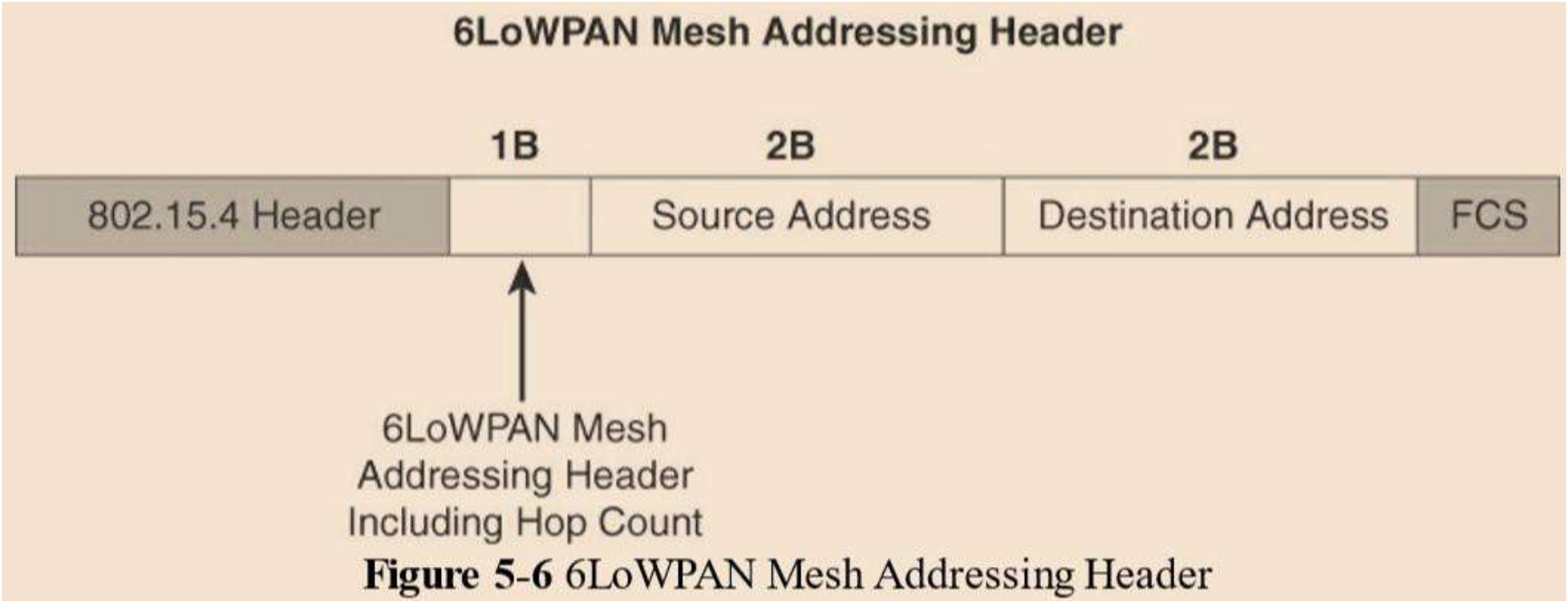


Figure 5-6 6LoWPAN Mesh Addressing Header

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **Mesh-Under Versus Mesh-Over Routing:**
  - IEEE 802.15.4, IEEE 802.15.4g, and IEEE 1901.2a that support mesh topologies and operate at the physical and data link layers, two main options exist for establishing reachability and forwarding packets.
    - "Mesh-under": the routing of packets is handled at the 6LoWPAN adaptation layer.
    - "Mesh-over" or "route-over": utilizes IP routing for getting packets to their destination.

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **Mesh-under routing,**
  - the routing of IP packets leverages the 6LoWPAN mesh addressing header to route and forward packets at the link layer.
  - The term mesh-under is used because multiple link layer hops can be used to complete a single IP hop.
  - Nodes have a Layer 2 forwarding table that they consult to route the packets to their final destination within the mesh.
  - An edge gateway terminates the mesh-under domain.
    - The edge gateway must also implement a mechanism to translate between the configured Layer 2 protocol and any IP routing mechanism implemented on other Layer 3 IP interfaces.

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **Mesh-over or route-over scenarios,**
  - IP Layer 3 routing is utilized for computing reachability and then getting packets forwarded to their destination, either inside or outside the mesh domain.
  - Each full-functioning node acts as an IP router, so each link layer hop is an IP hop.
  - When a LoWPAN has been implemented using different link layer technologies, a mesh-over routing setup is useful.
    - While traditional IP routing protocols can be used, a specialized routing protocol for smart objects, such as RPL

Optimizing IP for IoT From 6LoWPAN to 6Lo

- **6Lo Working Group:**
- The 6Lo working group seeks to expand on this completed work with a focus on IPv6 connectivity over constrained node networks.
    - While the 6LoWPAN working group initially focused its optimizations on IEEE 802.15.4 LLNs,

Optimizing IP for IoT From 6LoWPAN to 6Lo

- 6Lo working group is focused on the following:
- **IPv6-over-foo adaptation layer specifications using 6LoWPAN technologies (RFC4944, RFC6282, RFC6775) for link layer technologies:**
  - For example, this includes:
  - IPv6 over Bluetooth Low Energy
  - Transmission of IPv6 packets over near-field communication
  - IPv6 over 802.11ah
  - Transmission of IPv6 packets over DECT Ultra Low Energy
  - Transmission of IPv6 packets on WIA-PA (Wireless Networks for Industrial Automation–Process Automation)
  - Transmission of IPv6 over Master Slave/Token Passing (MS/TP)
- **Information and data models such as MIB modules:**
  - One example is RFC 7388, "Definition of Managed Objects for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)."
- **Optimizations that are applicable to more than one adaptation layer specification:**
  - For example, this includes RFC 7400, "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)."
- **Informational and maintenance publications needed for the IETF specifications in this area**

Optimizing IP for IoT 6TiSCH

- IEEE 802.15.4e, Time-Slotted Channel Hopping (TSCH), is an add-on to the Media Access Control (MAC) portion of the IEEE 802.15.4 standard Devices implementing IEEE 802.15.4e
- TSCH communicate by following a Time Division Multiple Access (TDMA) schedule.
    - An allocation of a unit of bandwidth or time slot is scheduled between neighbor nodes.
    - This allows the programming of predictable transmissions and enables deterministic, industrial-type applications.
    - Not like other IEEE 802.15.4

Optimizing IP for IoT 6TiSCH

- To standardize IPv6 over the TSCH mode of IEEE 802.15.4e (known as 6TiSCH)
- The IEEE 802.15.4e standard defines a time slot structure, but it does not mandate a scheduling algorithm for how the time slots are utilized.
    - This is left to higher-level protocols like 6TiSCH.
    - Scheduling is critical because it can affect throughput, latency, and power consumption.

Optimizing IP for IoT 6TiSCH

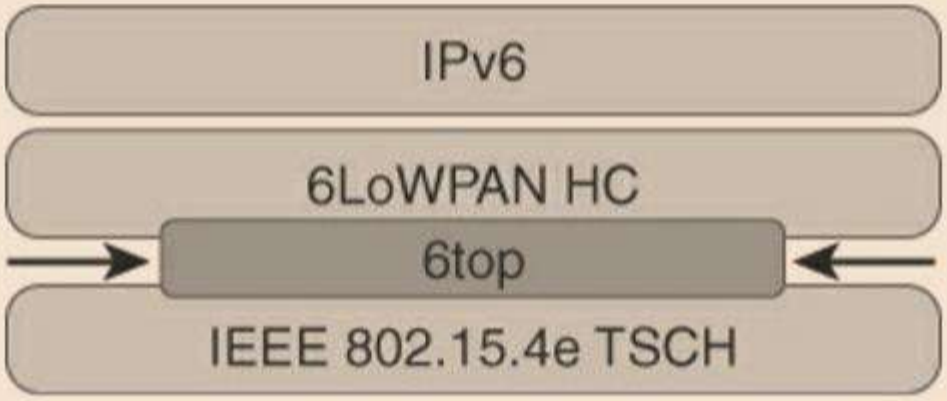


Figure 5-7 Location of 6TiSCH's 6top Sublayer

Optimizing IP for IoT 6TiSCH

- Schedules in 6TiSCH are broken down into cells.
  - A cell is simply a single element in the TSCH schedule that can be allocated for unidirectional or bidirectional communications between specific nodes.
  - Nodes only transmit when the schedule dictates that their cell is open for communication.
- The 6TiSCH architecture defines four schedule management mechanisms:
  - Static scheduling
  - Neighbor-to-neighbor scheduling
  - Remote monitoring and scheduling management
  - Hop-by-hop scheduling

Optimizing IP for IoT 6TiSCH

- **Static scheduling:**
- All nodes in the constrained network share a fixed schedule.
- Cells are shared, and nodes contend for slot access in a slotted aloha manner.
  - Slotted aloha is a basic protocol for sending data using time slot boundaries when communicating over a shared medium.
- Static scheduling is a simple scheduling mechanism that can be used upon initial implementation or as a fall back in the case of network malfunction.
- The drawback with static scheduling is that nodes may expect a packet at any cell in the schedule.
  - Therefore, energy is wasted idly listening across all cells.
- **Neighbor-to-neighbor scheduling:**
- A schedule is established that correlates with the observed number of transmissions between nodes.
- Cells in this schedule can be added or deleted as traffic requirements and bandwidth needs change.

Optimizing IP for IoT 6TiSCH

- **Remote monitoring and scheduling management:**
- Time slots and other resource allocation are handled by a management entity that can be multiple hops away.
- The scheduling mechanism leverages 6top and even CoAP in some scenarios.
- This scheduling mechanism provides quite a bit of flexibility and control in allocating cells for communication between nodes.
- **Hop-by-hop scheduling:**
- A node reserves a path to a destination node multiple hops away by requesting the allocation of cells in a schedule at each intermediate node hop in the path.
- The protocol that is used by a node to trigger this scheduling mechanism is not defined at this point.

Optimizing IP for IoT 6TiSCH

- In addition to schedule management functions, the 6TiSCH architecture also defines three different **forwarding models.**
- Forwarding is the operation performed on each packet by a node that allows it to be delivered to a next hop or an upper- layer protocol.
- The forwarding decision is based on a pre existing state that was learned from a routing computation.
- There are three 6TiSCH forwarding models:
  - Track Forwarding (TF)
  - Fragment forwarding (FF)
  - IPv6 Forwarding (6F)

Optimizing IP for IoT 6TiSCH

- **Track Forwarding (TF):**
  - This is the simplest and fastest forwarding model.
  - A “track” in this model is a unidirectional path between a source and a destination.
    - This track is constructed by pairing bundles of receive cells in a schedule with a bundle of receive cells set to transmit. So, a frame received within a particular cell or cell bundle is switched to another cell or cell bundle.
  - This forwarding occurs regardless of the network layer protocol.
- **IPv6 Forwarding (6F):**
  - This model forwards traffic based on its IPv6 routing table.
  - Flows of packets should be prioritized by traditional QoS (quality of service) and RED (random early detection) operations.
    - QoS is a classification scheme for flows based on their priority, and RED is a common congestion avoidance mechanism.

Optimizing IP for IoT 6TiSCH

- **Fragment forwarding (FF):**
  - This model takes advantage of 6LoWPAN fragmentation to build a Layer 2 forwarding table.
  - Fragmentation within the 6LoWPAN protocol.
  - IPv6 packets can get fragmented at the 6LoWPAN sublayer to handle the differences between IEEE 802.15.4 payload size and IPv6 MTU.
  - Additional headers for RPL source route information can further contribute to the need for fragmentation.
  - However, with FF, a mechanism is defined where the first fragment is routed based on the IPv6 header present.
  - The 6LoWPAN sublayer learns the next-hop selection of this first fragment, which is then applied to all subsequent fragments of that packet. Otherwise, IPv6 packets undergo hop-by-hop reassembly.
  - This increases latency and can be power- and CPU-intensive for a constrained node.

Optimizing IP for IoT RPL

- IETF chartered the RoLL (Routing over Low-Power and Lossy Networks) working group to evaluate all Layer 3 IP routing protocols and determine the needs and requirements for developing a routing solution for IP smart objects.
- The new routing protocol should be developed for use by IP smart objects is IPv6 Routing Protocol for Low Power and Lossy Networks (RPL).

- In an RPL network,
  - each node acts as a router and becomes part of a mesh network.
  - Routing is performed at the IP layer.
  - Each node examines every received IPv6 packet and determines the next-hop destination based on the information contained in the IPv6 header.

Optimizing IP for IoT RPL

- The constraints of computing and memory that are common characteristics of constrained nodes, the protocol defines two modes:

- **Storing mode:**
  - All nodes contain the full routing table of the RPL domain. Every node knows how to directly reach every other node.
- **Non-storing mode:**
  - Only the border router(s) of the RPL domain contain(s) the full routing table.
  - All other nodes in the domain only maintain their list of parents and use this as a list of default routes toward the border router.
  - This abbreviated routing table saves memory space and CPU.
  - When communicating in non-storing mode, a node always forwards its packets to the border router, which knows how to ultimately reach the final destination.

Optimizing IP for IoT RPL

- RPL is based on the concept of a directed acyclic graph (DAG). A DAG is a directed graph where no cycles exist. This means that from any vertex or point in the graph, you cannot follow an edge or a line back to this same point. All of the edges are arranged in paths oriented toward and terminating at one or more root nodes.

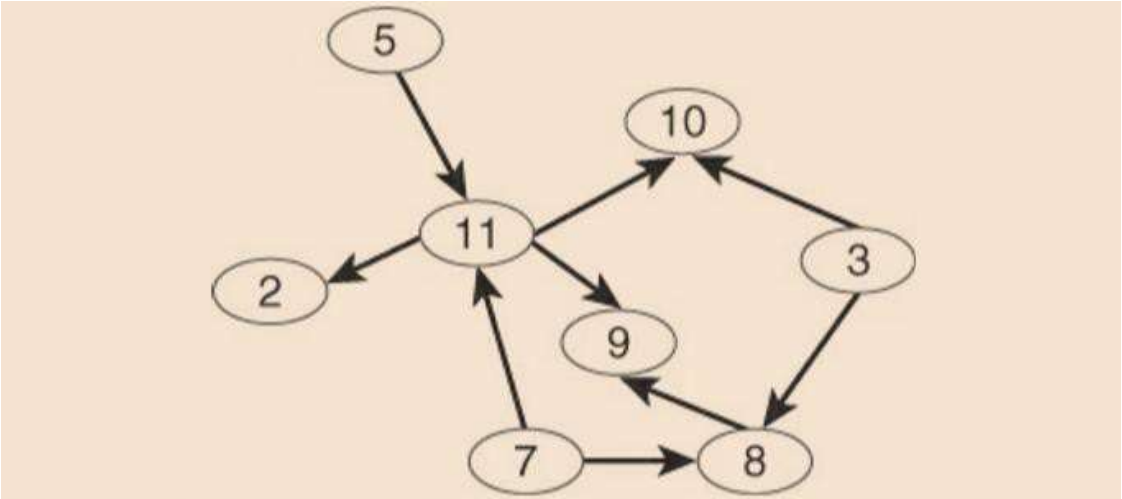


Figure 5-8 Example of a Directed Acyclic Graph (DAG)

Optimizing IP for IoT RPL

- A basic RPL process involves building a destination-oriented directed acyclic graph (DODAG).
- A DODAG is a DAG rooted to one destination.
- In RPL, this destination occurs at a border router known as the DODAG root.
- Observe that that a DAG has multiple roots, whereas the DODAG has just
- one.

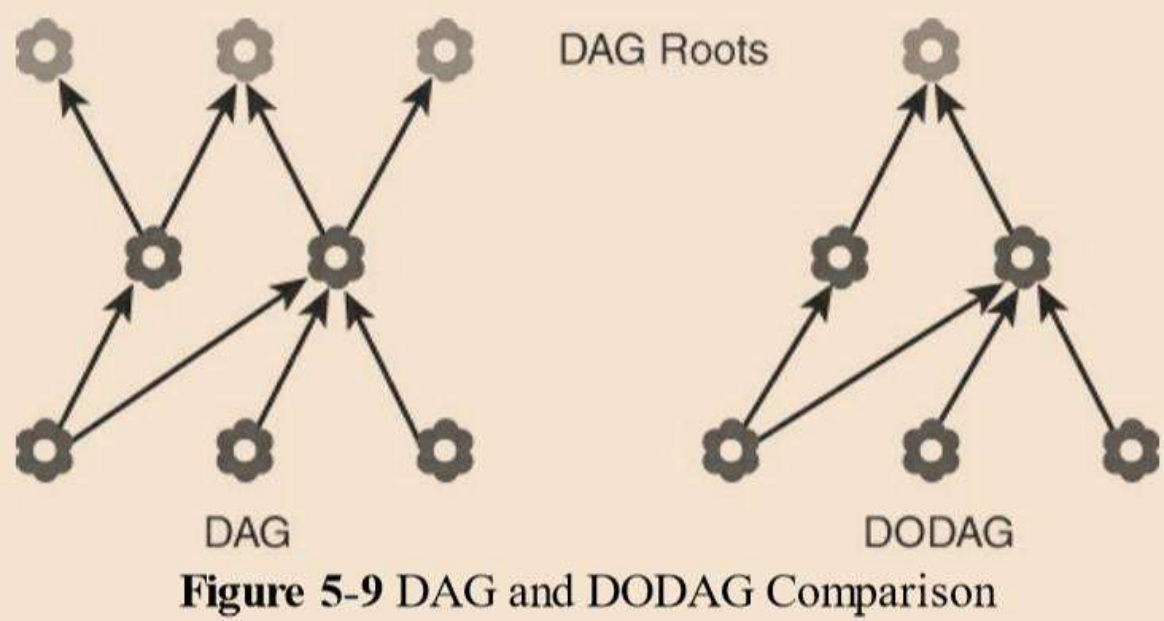


Figure 5-9 DAG and DODAG Comparison

Optimizing IP for IoT RPL

- In a DODAG, each node maintains up to three parents that provide a path to the root.
  - one of these parents is the preferred parent, which means it is the preferred next hop for upward routes toward the root.
- The routing graph created by the set of DODAG parents across all nodes defines the full set of upward routes.
- RPL protocol implementation should ensure that routes are loop free by disallowing nodes from selected DODAG parents that are positioned further away from the border router.
- Upward routes in RPL are discovered and configured using DAG Information Object (DIO) messages.
  - Nodes listen to DIOs to handle changes in the topology that can affect routing.

Optimizing IP for IoT RPL

- Nodes establish downward routes by advertising their parent set toward the DODAG root using a Destination Advertisement Object (DAO) message.
  - DAO messages allow nodes to inform their parents of their presence and reachability to descendants.
- In non-storing mode of RPL,
  - nodes sending DAO messages report their parent sets directly to the DODAG root (border router), and only the root stores the routing information.
- In storing mode,
  - each node keeps track of the routing information that is advertised in the DAO messages.
  - While this is more power- and CPU-intensive for each node, the benefit is that packets can take shorter paths between destinations in the mesh.
  - The nodes can make their own routing decisions; in non-storing mode, on the other hand, all packets must go up to the root to get a route for moving downstream.
- RPL messages, such as DIO and DAO, run on top of IPv6. These messages exchange and advertise downstream and upstream routing information between a border router and the nodes under it.

Optimizing IP for IoT RPL

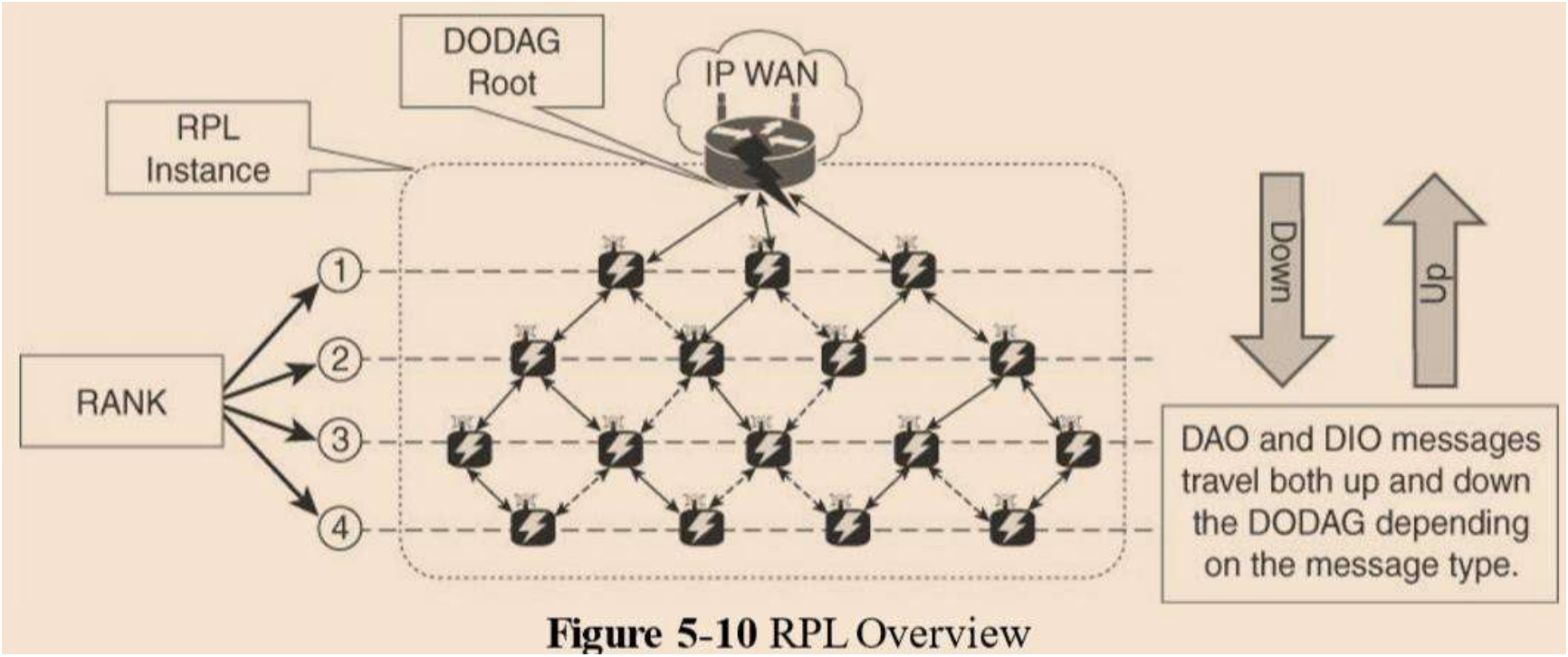


Figure 5-10 RPL Overview

Optimizing IP for IoT

RPL

- **Objective Function (OF) :**
  - An objective function (OF) defines how metrics are used to select routes and establish a node's rank.
  - For example, nodes implementing an OF with Minimum Expected Number of Transmissions (METX) advertise the METX among their parents in DIO messages.
  - Whenever a node establishes its rank, it simply sets the rank to the current minimum METX among its parents.
- **Rank**
  - The rank is a rough approximation of how "close" a node is to the root and helps avoid routing loops and the count-to-infinity problem.
  - Nodes can only increase their rank when receiving a DIO message with a larger version number.
  - However, nodes may decrease their rank whenever they have established lower-cost routes.
  - While the rank and routing metrics are closely related, the rank differs from routing metrics in that it is used as a constraint to prevent routing loops.

Optimizing IP for IoT

RPL

- **RPL Headers:**
- An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL).
- A new IPv6 option, known as the RPL option
  - The RPL option is carried in the IPv6 Hop-by-Hop header.
  - The purpose of this header is to leverage data plane packets for loop detection in a RPL instance.
- Source Routing Header (SRH) for use between RPL routers.
- A border router or DODAG root inserts the SRH when specifying a source route to deliver datagrams to nodes downstream in the mesh network.

Optimizing IP for IoT

RPL

- **Metrics**
- Some of the RPL routing metrics and constraints defined in RFC 6551 include the following:
- Expected Transmission Count (ETX):
  - Assigns a discrete value to the number of transmissions a node expects to make to deliver a packet.
- Hop Count:
  - Tracks the number of nodes traversed in a path. Typically, a path with a lower hop count is chosen over a path with a higher hop count.
- Latency:
  - Varies depending on power conservation. Paths with a lower latency are preferred.
- Link Quality Level:
  - Measures the reliability of a link by taking into account packet error rates caused by factors such as signal attenuation and interference.
- Link Color:
  - Allows manual influence of routing by administratively setting values to make a link more or less desirable. These values can be either statically or dynamically adjusted for specific traffic types.

Optimizing IP for IoT

RPL

- Node State and Attribute:
  - Identifies nodes that function as traffic aggregators and nodes that are being impacted by high workloads. High workloads could be indicative of nodes that have incurred high CPU or low memory states. Naturally, nodes that are aggregators are preferred over nodes experiencing high workloads.
- Node Energy:
  - Avoids nodes with low power, so a battery-powered node that is running out of energy can be avoided and the life of that node and the network can be prolonged.
- Throughput:
  - Provides the amount of throughput for a node link. Often, nodes conserving power use lower throughput. This metric allows the prioritization of paths with higher throughput.

Optimizing IP for IoT

Authentication and Encryption on Constrained Nodes

- The IETF working groups that are focused on IoT security:
  - ACE and DICE.

Optimizing IP for IoT

Authentication and Encryption on Constrained Nodes

- **ACE:**
  - The **Authentication and Authorization for Constrained Environments (ACE)** working group is tasked with evaluating the applicability of existing authentication and authorization protocols and documenting their suitability for certain constrained-environment use cases.
  - ACE working group will focus its work on CoAP with the Datagram Transport Layer Security (DTLS) protocol.
  - The ACE working group expects to produce a standardized solution for authentication and authorization that enables authorized access (Get, Put, Post, Delete) to resources identified by a URI and hosted on a resource server in constrained environments.
  - An unconstrained authorization server performs mediation of the access.

Optimizing IP for IoT

Authentication and Encryption on Constrained Nodes

- **DICE:**
  - New generations of constrained nodes implementing an IP stack over constrained access networks are expected to run an optimized IP protocol stack.
  - The **DTLS in Constrained Environments (DICE)** working group focuses on implementing the DTLS transport layer security protocol in these environments.
  - The first task of the DICE working group is to define an optimized DTLS profile for constrained nodes.
  - In addition, the DICE working group is considering the applicability of the DTLS record layer to secure multicast messages and investigating how the DTLS handshake in constrained environments can get optimized.

Profiles and Compliances

- Profile definitions, certifications, and promotion by alliances can help implementers develop solutions that guarantee interoperability and/or interchangeability of devices.
- Some of the main industry organizations working on profile definitions and certifications for IoT constrained nodes and networks.
  - Internet Protocol for Smart Objects (IPSO) Alliance
  - Wi-SUN Alliance
  - Thread
  - IPv6 Ready Logo

Profiles and Compliances

Internet Protocol for Smart Objects (IPSO) Alliance

- The alliance initially focused on promoting IP as the premier solution for smart objects communications.
- Today, it is more focused on how to use IP, with the IPSO Alliance organizing interoperability tests between alliance members to validate that IP for smart objects can work together and properly implement industry standards.

Profiles and Compliances Wi-SUN Alliance

- Wi-SUN's main focus is on the IEEE 802.15.4g protocol and its support for multiservice and secure IPv6 communications with applications running over the UDP transport layer.
- The utilities industry is the main area of focus for the Wi-SUN Alliance.
- The Wi-SUN field area network (FAN) profile enables smart utility networks to provide resilient, secure, and cost-effective connectivity with extremely good coverage in a range of topographic environments, from dense urban neighborhoods to rural areas.

Profiles and Compliances

Thread

- Thread Group has defined an IPv6-based wireless profile that provides the best way to connect more than 250 devices into a low-power, wireless mesh network.

Profiles and Compliances IPv6 Ready Logo

- The IPv6 Ready Logo program has established conformance and interoperability testing programs with the intent of increasing user confidence when implementing IPv6.
- The IPv6 Core and specific IPv6 components, such as DHCP, IPsec, and customer edge router certifications, are in place.

APPLICATION PROTOCOLS FOR IOT

Introduction

- Concepts covered about the higher-layer IoT protocols
  - **The Transport Layer:**
    - IP-based networks use either TCP or UDP. However, the constrained nature of IoT networks requires a closer look at the use of these traditional transport mechanisms.
  - **IoT Application Transport Methods:**
    - The various types of IoT application data and the ways this data can be carried across a network.

The Transport layer

- The selection of a protocol for the transport layer as supported by the TCP/IP architecture in the context of IoT networks.
- With the TCP/IP protocol, two main protocols are specified for the transport layer:
  - **Transmission Control Protocol (TCP):**
    - This connection-oriented protocol requires a session to get established between the source and destination before exchanging data.
  - **User Datagram Protocol (UDP):**
    - With this connectionless protocol, data can be quickly sent between source and destination but with no guarantee of delivery.

The Transport layer

- TCP is the main protocol used at the transport layer.
  - This is largely due to its inherent characteristics, such as its ability to transport large volumes of data into smaller sets of packets.
  - In addition, it ensures reassembly in a correct sequence, flow control and window adjustment, and retransmission of lost packets.
  - These benefits occur with the cost of overhead per packet and per session, potentially impacting overall packet per second performances and latency.
- UDP
  - is most often used in the context of network services, such as Domain Name System (DNS), Network Time Protocol (NTP), Simple Network Management Protocol (SNMP), and Dynamic Host Control Protocol (DHCP), or for real-time data traffic, including voice and video over IP.
  - In these cases, performance and latency are more important than packet retransmissions because re-sending a lost voice or video packet does not add value.
  - When the reception of packets must be guaranteed error free, the application layer protocol takes care of that function.
- When considering the choice of a transport layer by a given IoT application layer protocol, it is recommended to evaluate the impact of this choice on both the lower and upper layers of the stack.

The Transport layer

- Because of constrained nodes and network need to use new IoT application protocol, such as Constrained Application Protocol (CoAP), almost always uses UDP and why implementations of industrial application layer protocols may call for the optimization and adoption of the UDP transport layer if run over LLNs.
  - Select TCP for cellular networks because these networks are typically more robust and can handle the overhead.
  - For LLNs, where both the devices and network itself are usually constrained, UDP is a better choice and often mandatory.
- TCP and UDP are the two main choices at the transport layer for the TCP/IP protocol. The performance and scalability of IoT constrained devices and networks is impacted by which one of these is selected.

IoT Application Transport Methods

- Concepts Covered:
  - Application Layer Protocol Not Present
  - SCADA
    - A Little Background on SCADA
    - Adapting SCADA for IP
    - Tunneling Legacy SCADA over IP Networks
    - SCADA Protocol Translation
    - SCADA Transport over LLNs with MAP-T
  - Generic Web-Based Protocols
  - IoT Application Layer Protocols
    - CoAP
    - Message Queuing Telemetry Transport (MQTT)

IoT Application Transport Methods

- The following categories of IoT application protocols and their transport methods:
- **Application layer protocol not present:**
  - In this case, the data payload is directly transported on top of the lower layers. No application layer protocol is used.
- **Supervisory control and data acquisition (SCADA):**
  - SCADA is one of the most common industrial protocols in the world, but it was developed long before the days of IP, and it has been adapted for IP networks.
- **Generic web-based protocols:**
  - Generic protocols, such as Ethernet, Wi-Fi, and 4G/LTE, are found on many consumer- and enterprise-class IoT devices that communicate over non-constrained networks.
- **IoT application layer protocols:**
  - IoT application layer protocols are devised to run on constrained nodes with a small compute footprint and are well adapted to the network bandwidth constraints on cellular or satellite links or constrained 6LoWPAN networks. Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP), are two well-known examples of IoT application layer protocols.

IoT Application Transport Methods

Application Layer Protocol Not Present

- In Class 0 send or receive only a few bytes of data.
  - For many reasons, such as processing capability, power constraints, and cost, these devices do not implement a fully structured network protocol stack, such as IP, TCP, or UDP, or even an application layer protocol.
  - Class 0 devices are usually simple smart objects that are severely constrained.
  - Implementing a robust protocol stack is usually not useful and sometimes not even possible with the limited available resources.
- While many constrained devices, such as sensors and actuators, have adopted deployments that have no application layer, this transportation method has not been standardized.
  - This lack of standardization makes it difficult for generic implementations of this transport method to be successful from an interoperability perspective.
  - E.g. Different kinds of temperature sensors from different manufacturers. These sensors will report temperature data in varying formats.
- The solution to this problem is to use an IoT data broker

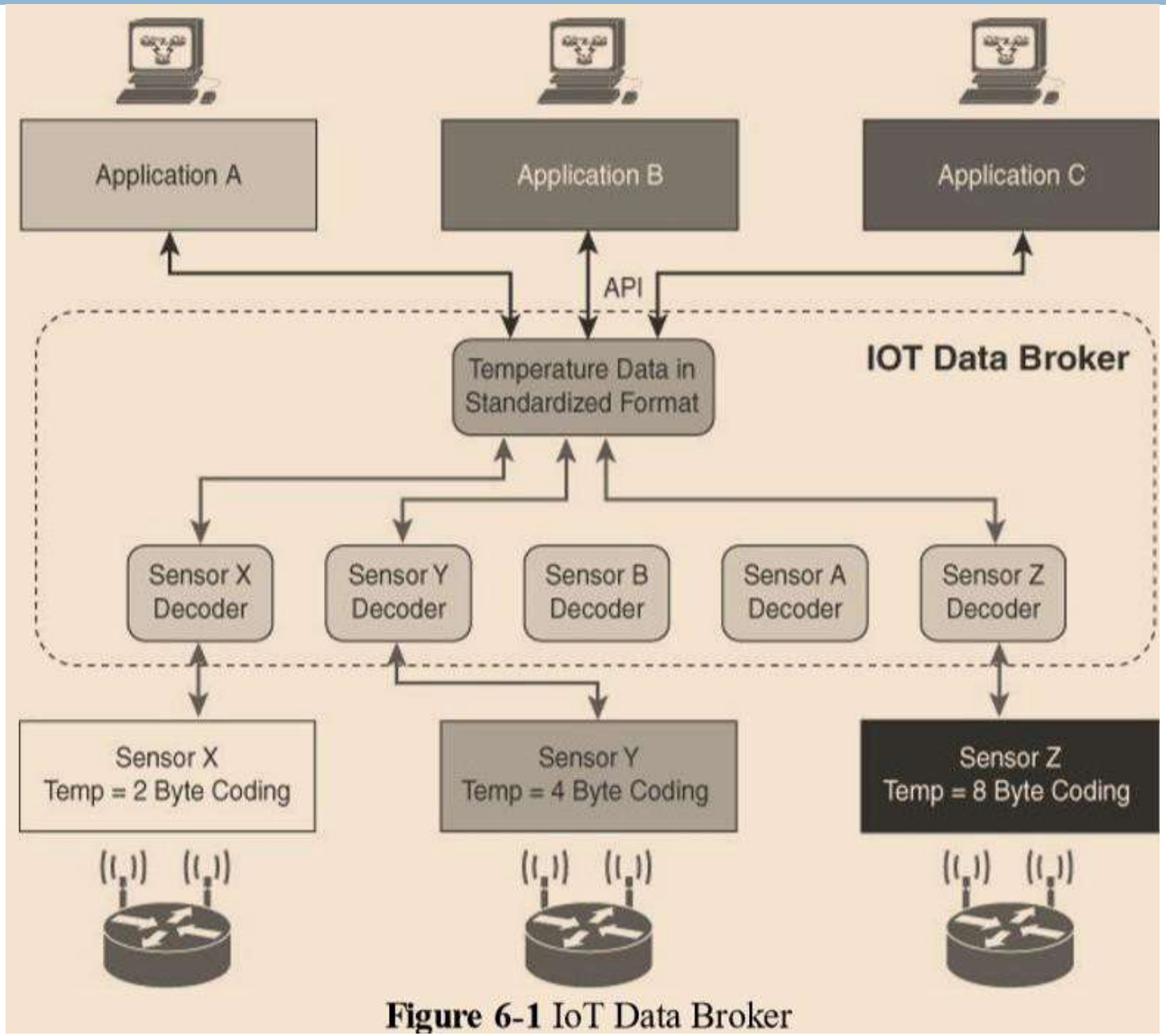


Figure 6-1 IoT Data Broker

IoT Application Transport Methods

Application Layer Protocol Not Present

- An IoT data broker is a piece of middleware that standardizes sensor output into a common format that can then be retrieved by authorized applications.
- In figure Sensors X, Y , and Z are all temperature sensors, but their output is encoded differently.
- The IoT data broker understands the different formats in which the temperature is encoded and is therefore able to decode this data into a common, standardized format.
- Applications A, B, and C in Figure can access this temperature data without having to deal with decoding multiple temperature data formats.

IoT Application Transport Methods

SCADA

- Supervisory control and data acquisition (SCADA).
- Designed decades ago, SCADA is an automation control system that was initially implemented without IP over serial links (such as RS-232 and RS-485), before being adapted to Ethernet and IPv4.

IoT Application Transport Methods SCADA - A little Background on SCADA

- SCADA networking protocols, running directly over serial physical and data link layers.
- At a high level, SCADA systems collect sensor data and telemetry from remote devices, and to control them.
- SCADA systems allow global, real-time, data-driven decisions to be made about how to improve business processes.
- SCADA commonly uses certain protocols for communications between devices and applications.
  - E.g.
  - Modbus industrial protocols used to monitor and program remote devices via a master/slave relationship.
    - Modbus used in building management, transportation, and energy applications.
  - The DNP3 (Distributed Network Protocol) and International Electrotechnical Commission (IEC) protocols are found mainly in the utilities industry, along with DLMS/COSEM
  - ANSI C12 for advanced meter reading (AMR).
- These protocols used decade ago and are serial based. So, transporting them over current IoT and traditional networks requires that certain Adjustments

IoT Application Transport Methods
SCADA - Adapting SCADA for IP

- Ethernet and IP include the ability to leverage existing equipment and standards while integrating seamlessly the SCADA subnetworks to the corporate WAN infrastructures.
- Assigning TCP/UDP to the protocols, as following:
  - DNP3 (adopted by IEEE 1815-2012) specifies the use of TCP or UDP on The Modbus messaging service utilizes TCP.
  - IEC60870-5-104 is the evolution of IEC60870-5-101 serial for running over Ethernet and IPv4 using port 2404.
  - DLMS User Association specified a communication profile based on TCP/IP in the DLMS/COSEM.

IoT Application Transport Methods
SCADA - Adapting SCADA for IP

- Serial protocols have adapted and evolved to utilize IP and TCP/UDP as both networking and transport mechanisms.
- DNP3 (Distributed Network Protocol) is based on a master/slave relationship.
  - The term master is refers to a powerful computer located in the control center of a utility, and a slave is a remote device with computing resources found in a location such as a substation.
  - DNP3 refers to slaves as outstations.
  - Outstations monitor and collect data from devices that indicate their state, such as whether a circuit breaker is on or off, and take measurements, including voltage, current, temperature, and so on.
  - This data is then transmitted to the master when it is requested, or events or alarms and control commands can be sent in an asynchronous manner.

IoT Application Transport Methods
SCADA - Adapting SCADA for IP

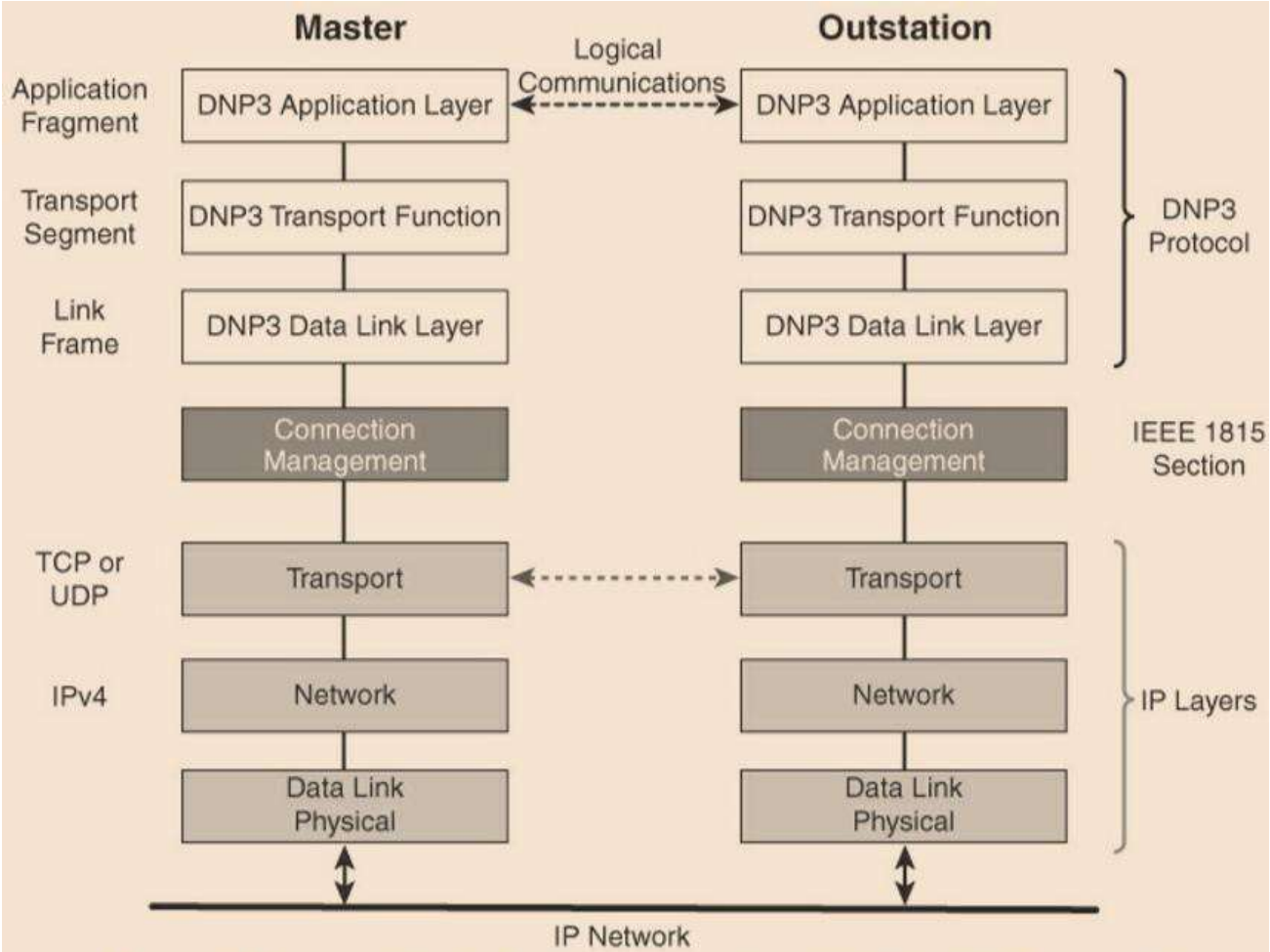


Figure 6-2 Protocol Stack for Transporting Serial DNP3 SCADA over IP

IoT Application Transport Methods

SCADA - Adapting SCADA for IP

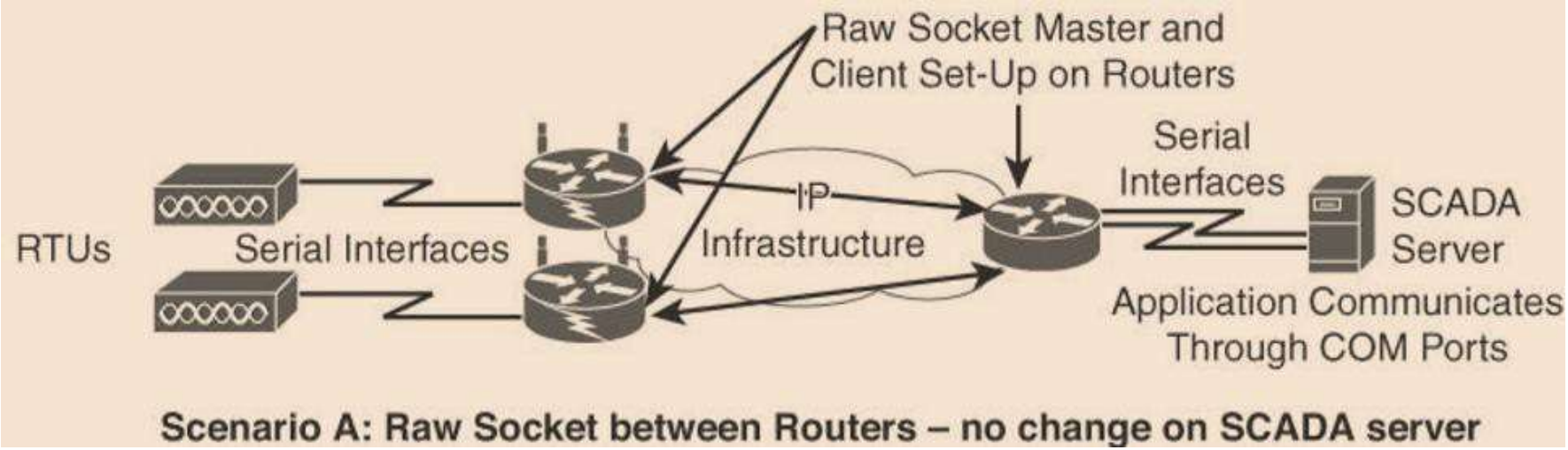
- Connection management links the DNP3 layers with the IP layers in addition to the configuration parameters and methods necessary for implementing the network connection.
- The master side initiates connections by performing a TCP active open.
- The outstation listens for a connection request by performing a TCP passive open.
- Master stations may parse multiple DNP3 data link layer frames from a single UDP datagram, while DNP3 data link layer frames cannot span multiple UDP datagrams.
- Single or multiple connections to the master may get established while a TCP keep alive timer monitors the status of the connection.

IoT Application Transport Methods SCADA - Tunneling Legacy SCADA over IP Networks

- End-to-end native IP support is preferred, in the case of DNP3.
- Otherwise, transport of the original serial protocol over IP can be achieved either by tunneling using raw sockets over TCP or UDP or by installing an intermediate device that performs protocol translation between the serial protocol version and its IP implementation.
- **A raw socket connection simply denotes that the serial data is being packaged directly into a TCP or UDP transport.**
- A socket is a standard application programming interface (API) composed of an IP address and a TCP or UDP port that is used to access network devices over an IP network.

IoT Application Transport Methods SCADA -

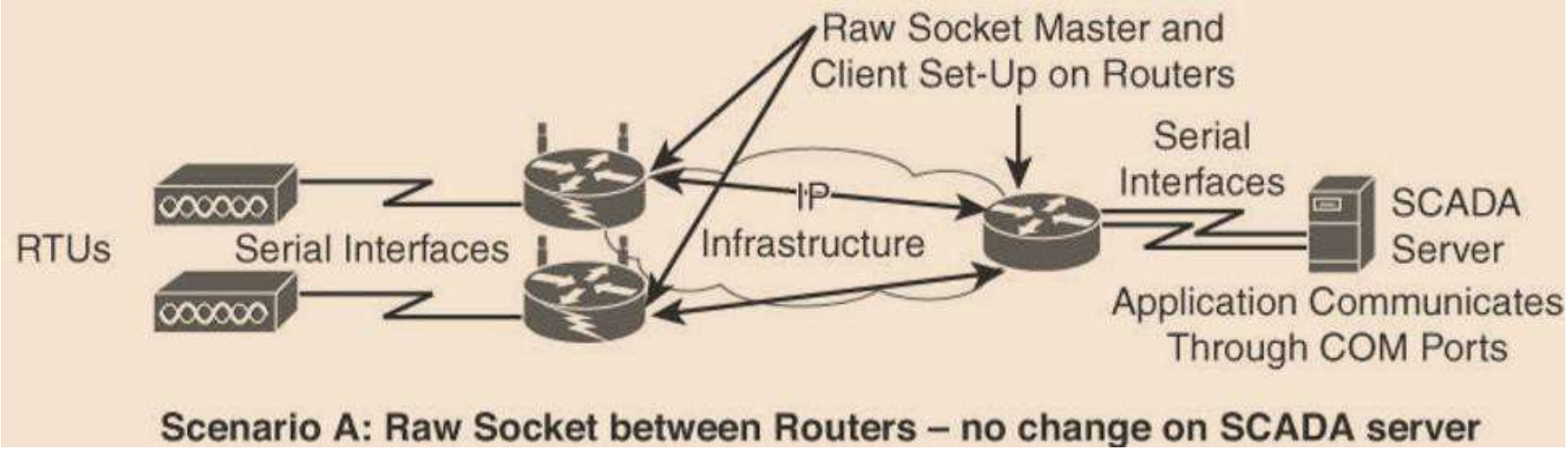
Tunneling Legacy SCADA over IP Networks



- Scenarios A, B and C in Figure,
- Routers connect via serial interfaces to the remote terminal units (RTUs), which are often associated with SCADA networks.
  - An RTU is a multipurpose device used to monitor and control various systems, applications, and devices managing automation.
  - From the master/slave perspective, the RTUs are the slaves.
- Opposite the RTUs in is a SCADA server, or master, that varies its connection type.

IoT Application Transport Methods SCADA -

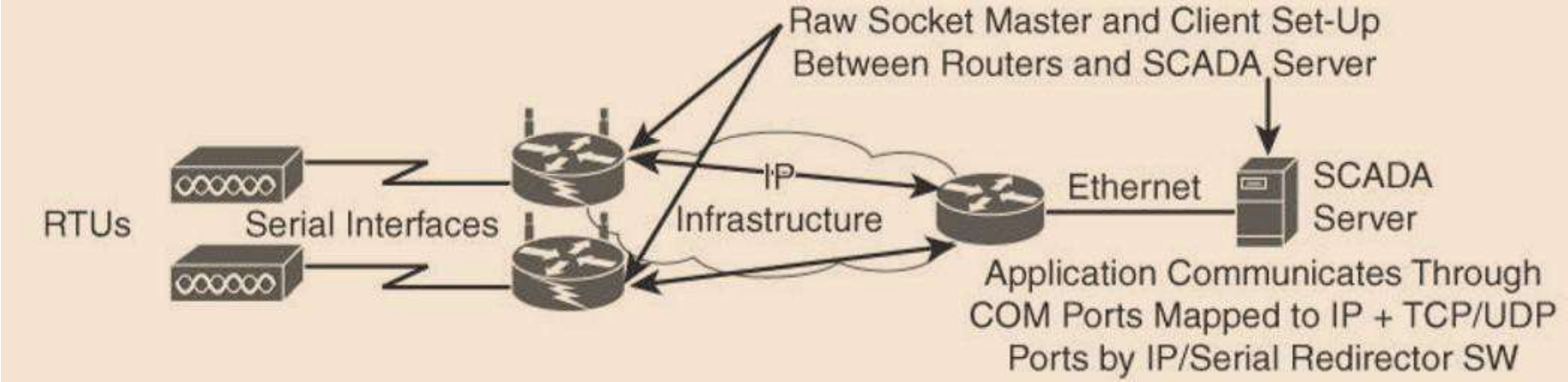
Tunneling Legacy SCADA over IP Networks



- Scenarios A:
  - Both the SCADA server and the RTUs have a direct serial connection to their respective routers.
  - The routers terminate the serial connections at both ends of the link and use raw socket encapsulation to transport the serial payload over the IP network.

IoT Application Transport Methods SCADA -

Tunneling Legacy SCADA over IP Networks

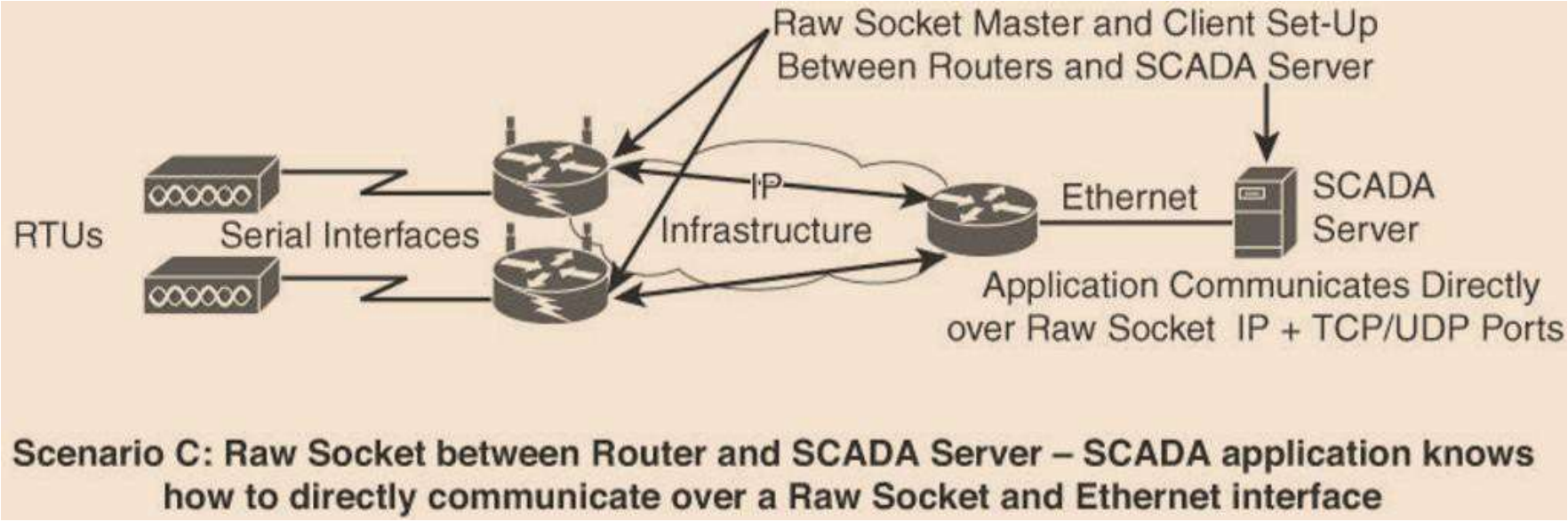


Scenario B: Raw Socket between Router and SCADA Server – no SCADA application change on server but IP/Serial Redirector software and Ethernet interface to be added

- Scenarios B:
  - A small change on the SCADA server side. A piece of software is installed on the SCADA server that maps the serial COM ports to IP ports. This software is commonly referred to as an IP/serial redirector. The IP/serial redirector in essence terminates the serial connection of the SCADA server and converts it to a TCP/IP port using a raw socket connection.

IoT Application Transport Methods SCADA -

Tunneling Legacy SCADA over IP Networks



Scenario C: Raw Socket between Router and SCADA Server – SCADA application knows how to directly communicate over a Raw Socket and Ethernet interface

- Scenarios C:
  - The SCADA server supports native raw socket capability.
  - Unlike in Scenarios A and B, where a router or IP/serial redirector software has to map the SCADA server's serial ports to IP ports, in Scenario C the SCADA server has full IP support for raw socket connections.

IoT Application Transport Methods SCADA - SCADA Protocol Translation

- With protocol translation, the legacy serial protocol is translated to a corresponding IP version.

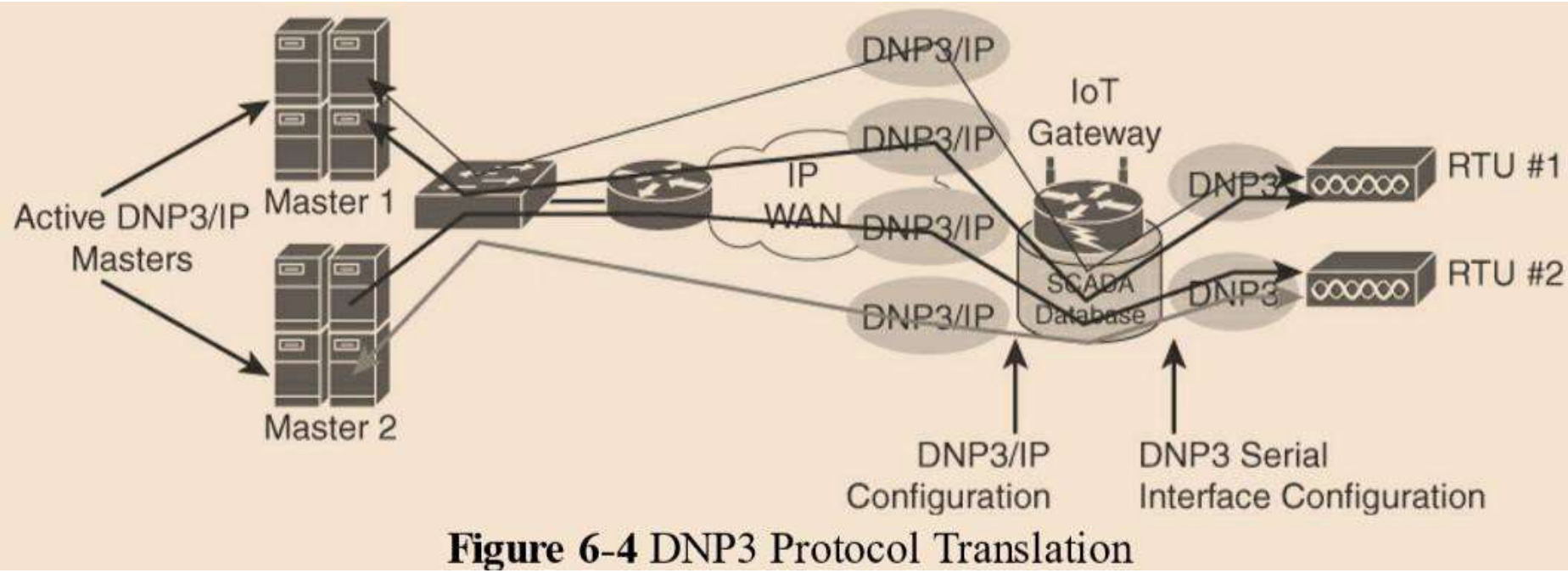


Figure 6-4 DNP3 Protocol Translation

IoT Application Transport Methods SCADA - SCADA Protocol Translation

- Figure shows two serially connected DNP3 RTUs and two master applications supporting DNP3 over IP that control and pull data from the RTUs.
- **The IoT gateway in this figure performs a protocol translation function that enables communication between the RTUs and servers, despite the fact that a serial connection is present on one side and an IP connection is used on the other.**
- By running protocol translation, the IoT gateway connected to the RTUs is implementing a computing function close to the edge of the network. Adding computing functions close to the edge helps scale distributed intelligence in IoT networks.

IoT Application Transport Methods

SCADA - SCADA Transport over LLNs with MAP-T

- Due to the constrained nature of LLNs, the implementation of industrial protocols should at a minimum be done over UDP.
- This in turn requires that both the application servers and devices support and implement UDP.
- When deployed over LLN subnetworks that are IPv6 only, a transition mechanism, such as **MAP-T (Mapping of Address and Port using Translation),** needs to be implemented.

IoT Application Transport Methods

SCADA - SCADA Transport over LLNs with MAP-T

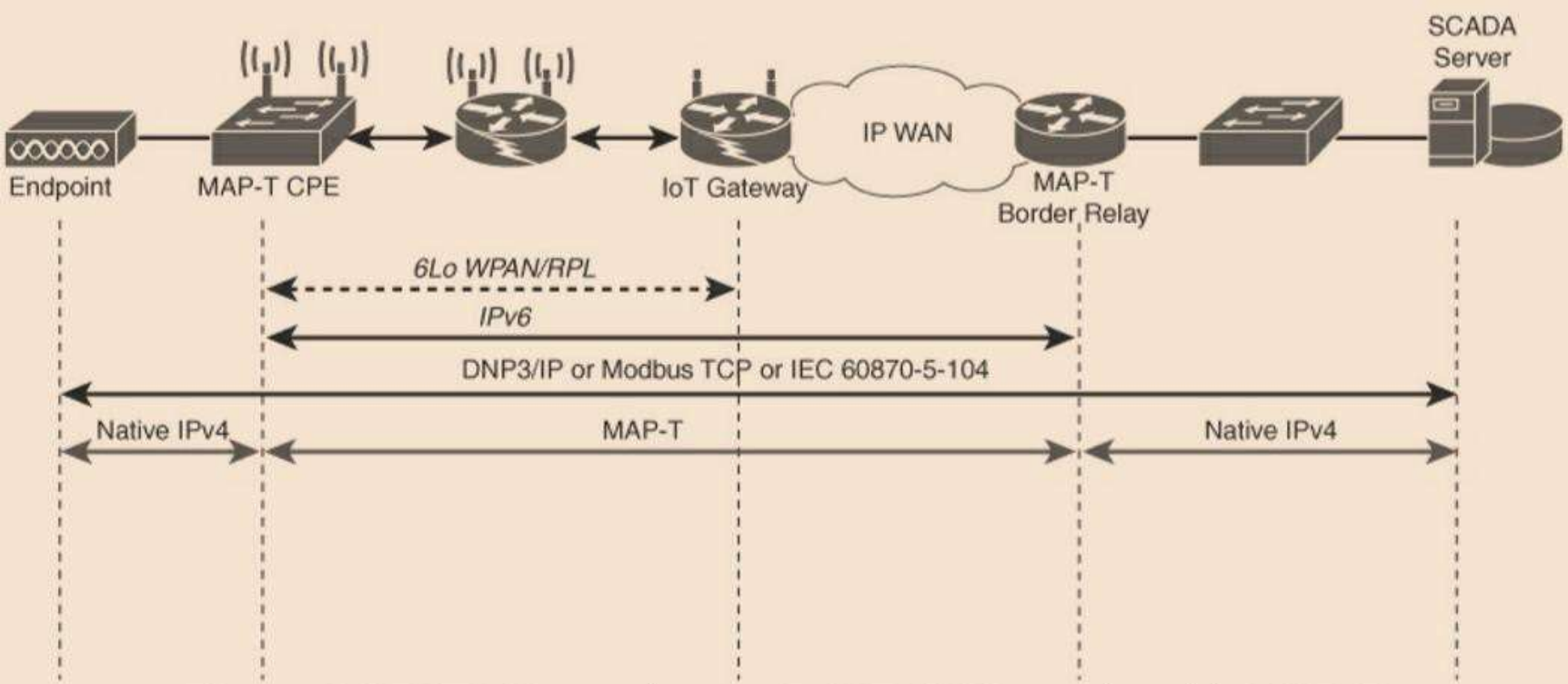


Figure 6-5 DNP3 Protocol over 6LoWPAN Networks with MAP-T

IoT Application Transport Methods

SCADA - SCADA Transport over LLNs with MAP-T

- In figure shows a scenario in which a legacy endpoint is connected across an LLN running 6LoWPAN to an IP-capable SCADA server.
- The legacy endpoint could be running various industrial and SCADA protocols, including DNP3/IP, Modbus/TCP, or IEC.
- In this scenario, the legacy devices and the SCADA server support only IPv4.
- MAP-T makes the appropriate mappings between IPv4 and the IPv6 protocols.
- This allows legacy IPv4 traffic to be forwarded across IPv6 networks.

- In Figure the IPv4 endpoint on the left side is connected to a Customer Premise Equipment (CPE) device.
- The MAP-T CPE device has an IPv6 connection to the RPL mesh.
- On the right side, a SCADA server with native IPv4 support connects to a MAP-T border gateway.

Generic Web-Based Protocols

- The level of familiarity with generic web-based protocols is high.
- Therefore, programmers with basic web programming skills can work on IoT applications, and this may lead to innovative ways to deliver and handle real-time IoT data.
- On **non-constrained networks, such as Ethernet, Wi-Fi, or 3G/4G** cellular, where bandwidth is not perceived as a potential issue, data payloads based on a verbose data model representation
- In case of **constrained networks the embedded web server software** with advanced features are now implemented with very little memory (in the range of 10KB).

Generic Web-Based Protocols

- IoT devices that only push data to an application may need to implement web services on the client side.
  - For example,
    - an Ethernet- or Wi-Fi-based weather station reporting data to a weather map application.
  - The HTTP client side only initiates connections and does not accept incoming ones.
- Some IoT devices, such as a video surveillance camera, may have web services implemented on the server side.
- Interactions between real-time communication tools powering collaborative applications, such as voice and video, instant messaging, chat rooms, and IoT devices, are also emerging.
  - Extensible Messaging and Presence Protocol (XMPP).

IoT Application layer Protocols

- When considering constrained networks and/or a large-scale deployment of constrained nodes, verbose web-based and data model protocols, may be too heavy for IoT applications.
- To address this problem, the new lightweight protocols that are better suited to large numbers of constrained nodes and networks.
- Two of the most popular protocols are
  - CoAP
  - MQTT

IoT Application Layer Protocols

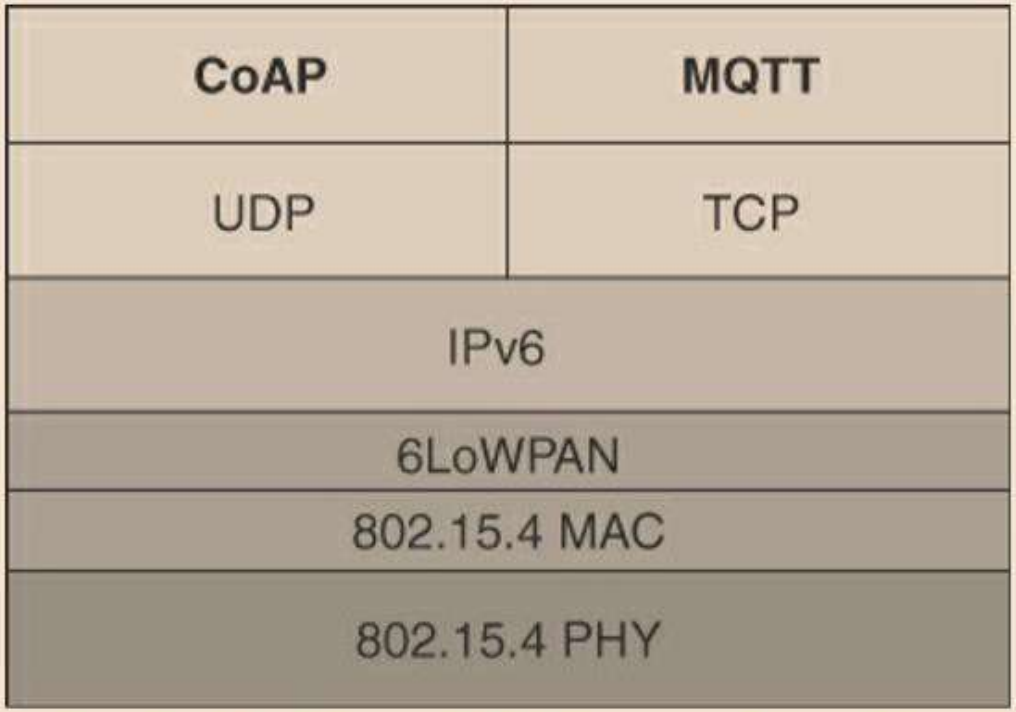


Figure 6-6 Example of a High-Level IoT Protocol Stack for CoAP and MQTT

IoT Application Layer Protocols

- CoAP and MQTT are at the top of this sample IoT stack, based on an IEEE 802.15.4 mesh network.
- CoAP deployed over UDP and MQTT running over TCP.

IoT Application layer Protocols CoAP

- **Constrained Application Protocol (CoAP)**
  - is to develop a generic framework for resource-oriented applications targeting constrained nodes and networks.
  - The CoAP framework defines simple and flexible ways to manipulate sensors and actuators for data or device management.
  - The CoAP messaging model is primarily designed to facilitate the exchange of messages over UDP between endpoints, including the **secure transport protocol Datagram Transport Layer Security (DTLS).**
  - CoAP over Short Message Service (SMS) as defined in Open Mobile Alliance for Lightweight Machine-to-Machine (LWM2M) for IoT device management.

IoT Application layer Protocols CoAP

- A CoAP message is composed of
  - a short fixed-length Header field (4 bytes),
  - a variable-length but mandatory Token field (0–8 bytes),
  - Options fields if necessary, and the Payload field.

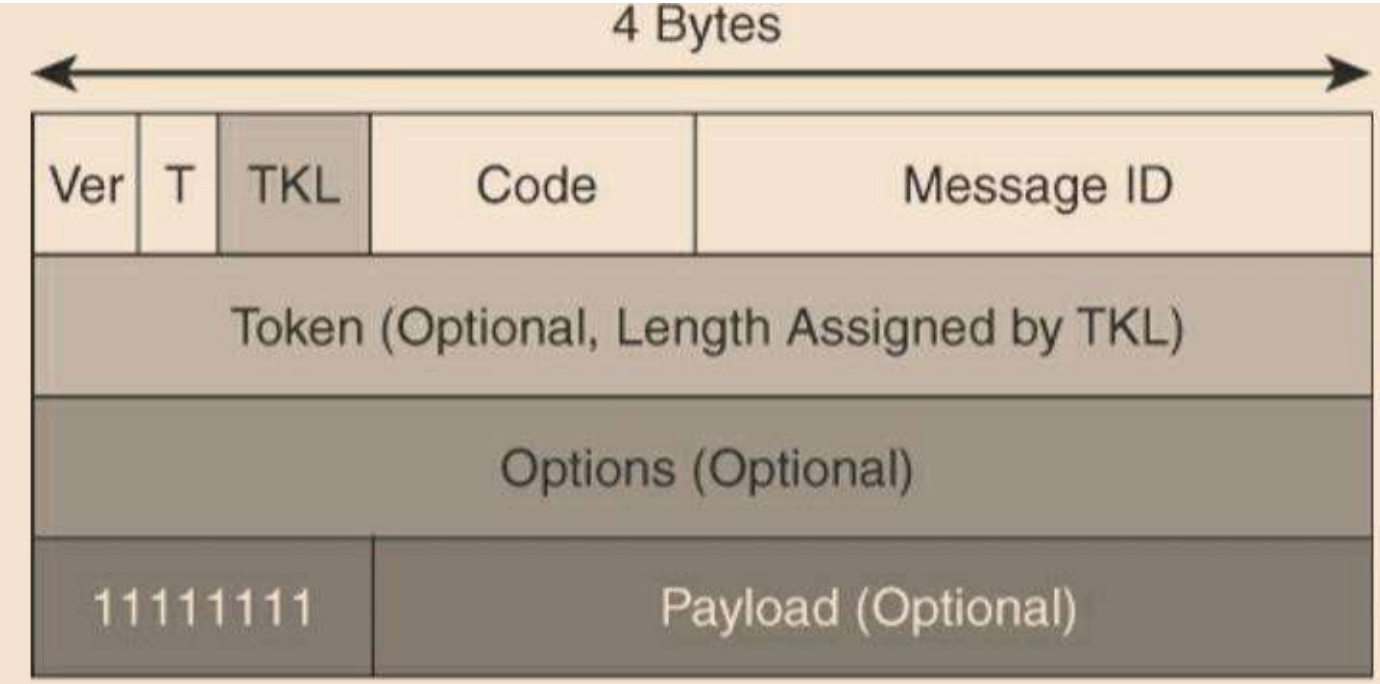


Figure 6-7 CoAP Message Format

IoT Application layer Protocols CoAP

CoAP Message Field	Description
Ver (Version)	Identifies the CoAP version.
T (Type)	Defines one of the following four message types: Confirmable (CON), Non-confirmable (NON), Acknowledgement (ACK), or Reset (RST). CON and ACK are highlighted in more detail in Figure 6-9.
TKL (Token Length)	Specifies the size (0–8 Bytes) of the Token field.
Code	Indicates the request method for a request message and a response code for a response message. For example, in Figure 6-9, GET is the request method, and 2.05 is the response code. For a complete list of values for this field, refer to RFC 7252.
Message ID	Detects message duplication and used to match ACK and RST message types to Con and NON message types.
Token	With a length specified by TKL, correlates requests and responses.
Options	Specifies option number, length, and option value. Capabilities provided by the Options field include specifying the target resource of a request and proxy functions.
Payload	Carries the CoAP application data. This field is optional, but when it is present, a single byte of all 1s (0xFF) precedes the payload. The purpose of this byte is to delineate the end of the Options field and the beginning of Payload.

Table 6-1 CoAP Message Fields

IoT Application layer Protocols CoAP

- CoAP can run over IPv4 or IPv6. However, it is recommended that the message fit within a single IP packet and UDP payload to avoid fragmentation.

IoT Application layer Protocols CoAP

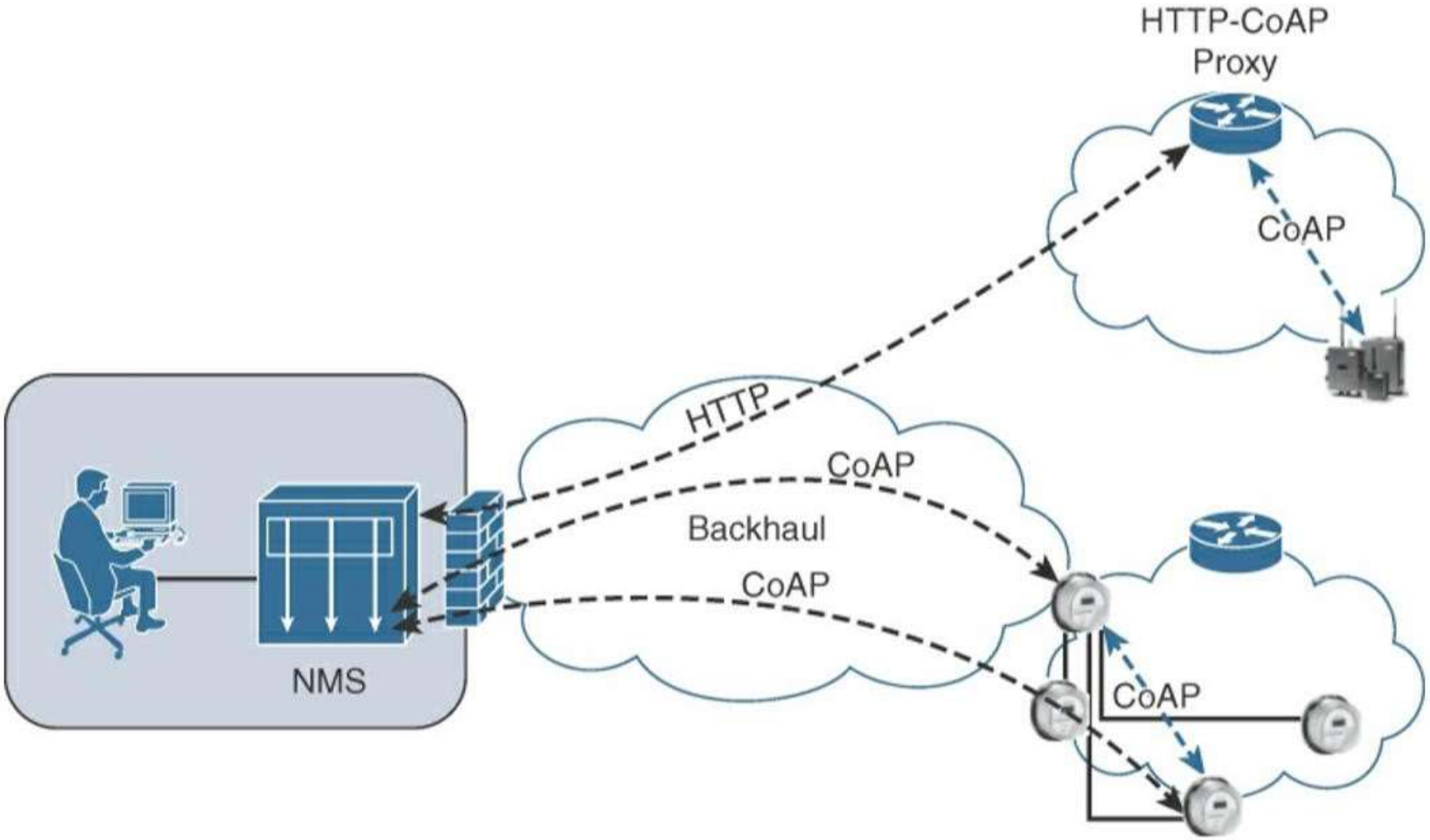


Figure 6-8 CoAP Communications in IoT Infrastructures

IoT Application layer Protocols CoAP

- CoAP communications across an IoT infrastructure can take various paths.
- Connections can be between devices located on the same or different constrained networks or between devices and generic Internet or cloud servers, all operating over IP.
- As both HTTP and CoAP are IP-based protocols, the proxy function can be located practically anywhere in the network, not necessarily at the border between constrained and non-constrained networks.
- Just like HTTP, CoAP is based on the REST architecture, but with a "thing" acting as both the client and the server.
  - Through the exchange of asynchronous messages, a client requests an action via a method code on a server resource.
  - A uniform resource identifier (URI) localized on the server identifies this resource.
  - The server responds with a response code that may include a resource representation.
  - The CoAP request/response semantics include the methods GET, POST, PUT, and DELETE.

IoT Application layer Protocols

MQTT

- **Message Queuing Telemetry Transport (MQTT) :**
  - Considering the harsh environments in the oil and gas industries, an extremely simple protocol with only a few options was designed, with considerations for constrained nodes, unreliable WAN backhaul communications, and bandwidth constraints with variable latencies.
  - These were some of the rationales for the selection of a client/server and publish/subscribe framework based on the TCP/IP architecture,

IoT Application layer Protocols

MQTT

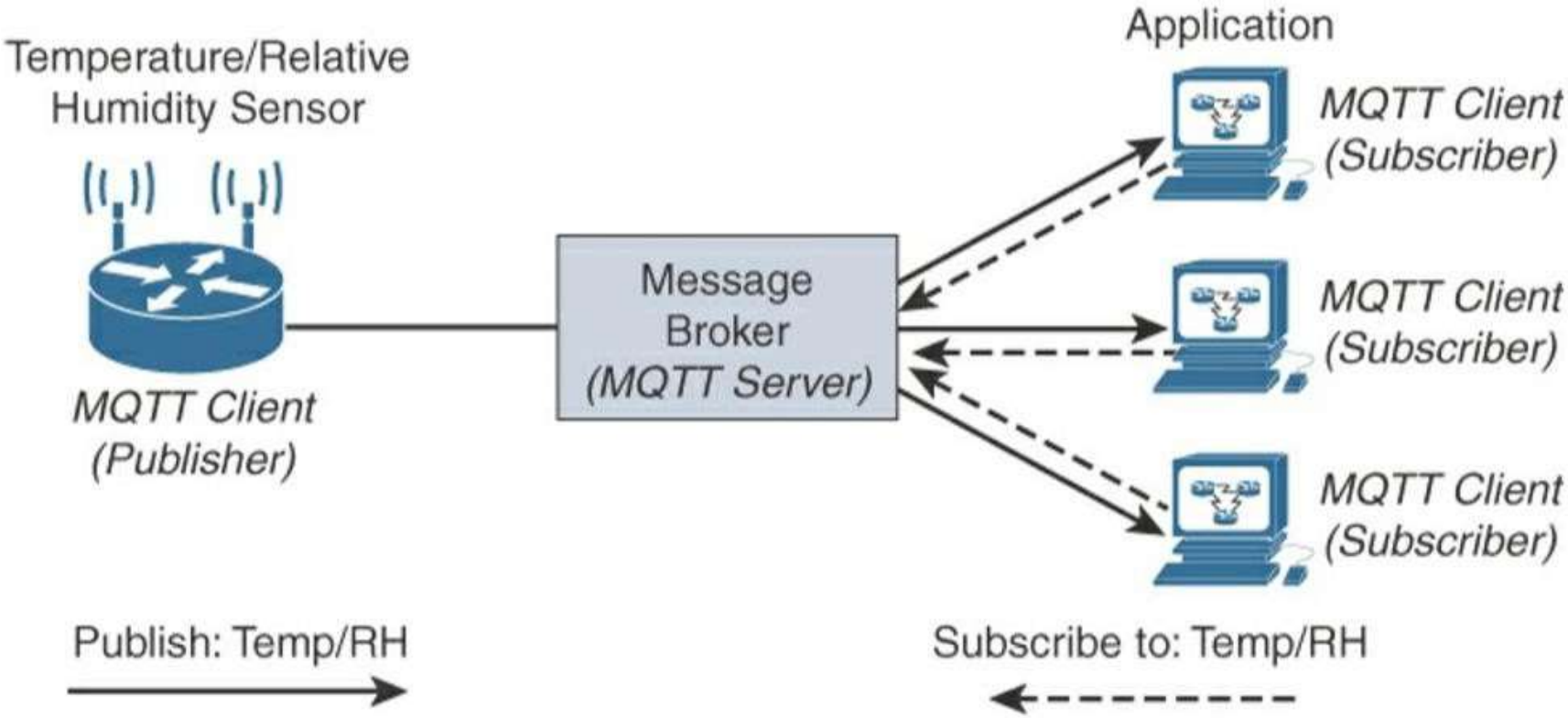


Figure 6-10 MQTT Publish/Subscribe Framework

IoT Application layer Protocols

MQTT

- An MQTT client can act as a publisher to send data (or resource information) to an MQTT server acting as an MQTT message broker.
  - In Figure the MQTT client on the left side is a temperature (Temp) and relative humidity (RH) sensor that publishes its Temp/RH data.
    - The MQTT server (or message broker) accepts the network connection along with application messages, such as Temp/RH data, from the publishers.
    - It also handles the subscription and unsubscription process and pushes the application data to MQTT clients acting as subscribers.
  - The application on the right side of Figure is an MQTT client that is a subscriber to the Temp/RH data being generated by the publisher or sensor on the left.
  - This model, where subscribers express a desire to receive information from publishers, is well known.

IoT Application layer Protocols

MQTT

- The presence of a message broker in MQTT decouples the data transmission between clients acting as publishers and subscribers.
  - In fact, publishers and subscribers do not even know (or need to know) about each other.
  - A benefit of having this decoupling is that the MQTT message broker ensures that information can be buffered and cached in case of network failures.

- Compared to the CoAP message, MQTT contains a smaller header of 2 bytes compared to 4 bytes for CoAP.

IoT Application layer Protocols

MQTT

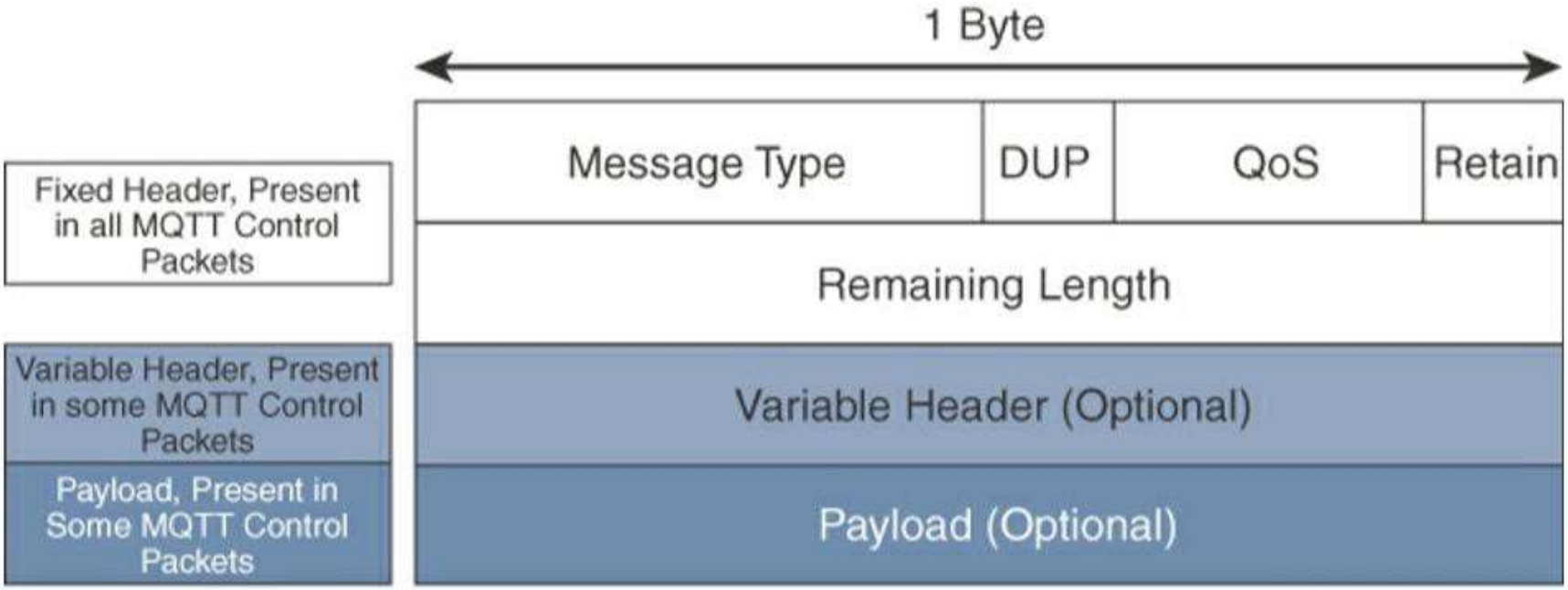


Figure 6-11 MQTT Message Format

IoT Application layer Protocols

MQTT

- MQTT is a lightweight protocol because each control packet consists of a 2-byte fixed header with optional variable header fields and optional payload.
- The first MQTT field in the header is Message Type, which identifies the kind of MQTT packet within a message.
  - Fourteen different types of control packets are specified in MQTT.
- Each of them has a unique value that is coded into the Message Type field.

IoT Application layer Protocols

MQTT

Message Type	Value	Flow	Description
CONNECT	1	Client to server	Request to connect
CONNACK	2	Server to client	Connect acknowledgement
PUBLISH	3	Client to server Server to client	Publish message
PUBACK	4	Client to server Server to client	Publish acknowledgement
PUBREC	5	Client to server Server to client	Publish received
PUBREL	6	Client to server Server to client	Publish release
PUBCOMP	7	Client to server Server to client	Publish complete
SUBSCRIBE	8	Client to server	Subscribe request
SUBACK	9	Server to client	Subscribe acknowledgement
UNSUBSCRIBE	10	Client to server	Unsubscribe request
UNSUBACK	11	Server to client	Unsubscribe acknowledgement
PINGREQ	12	Client to server	Ping request
PINGRESP	13	Server to client	Ping response
DISCONNECT	14	Client to server	Client disconnecting

Table 6-2 MQTT Message Types

IoT Application layer Protocols

MQTT

- The next field in the MQTT header is DUP (Duplication Flag).
  - This flag, when set, allows the client to notate that the packet has been sent previously, but an acknowledgement was not received.
- The QoS header field allows for the selection of three different QoS levels.
- The next field is the Retain flag.
  - Only found in a PUBLISH message, the Retain flag notifies the server to hold onto the message data.
  - This allows new subscribers to instantly receive the last known value without having to wait for the next update from the publisher.
- The last mandatory field in the MQTT message header is Remaining Length.
  - This field specifies the number of bytes in the MQTT packet following this field.

IoT Application layer Protocols

MQTT

- MQTT sessions between each client and server consist of four phases: session establishment, authentication, data exchange, and session termination.
- Each client connecting to a server has a unique client ID, which allows the identification of the MQTT session between both parties.
- When the server is delivering an application message to more than one client, each client is treated independently.
- Subscriptions to resources generate SUBSCRIBE/SUBACK control packets, while unsubscription is performed through the exchange of UNSUBSCRIBE/UNSUBACK control packets.
- Graceful termination of a connection is done through a DISCONNECT control packet, which also offers the capability for a client to reconnect by re-sending its client ID to resume the operations.